Weakly Supervised
Natural Language Understanding

Ni Lao
mosaix.ai
11.11.2018

For completeness a large part of the tutorial is from previous works. Thanks to Chen Liang for helping with creating these slides.
Speaker Background

- **BS from Tsinghua U. EE (1999 - 2003)** Worked on the TsinghuAeolus system, and won world champion in RoboCup simulation league in 2001 and 2002

- **MS from Tsinghua U. CS (2003 - 2006)** Worked at Microsoft Research Asia on automatic OS diagnosis, Web search and product search

- **PhD from Carnegie Mellon U. (2006 - 2012)** Researched on IR, ML, NLP. Worked on the CMU JAVELIN QA system, and Never-Ending Learning (NELL) system


- **Chief Scientist & Co-founder Mosaix.ai (2018 - )** Research on semantic parsing and text understanding for search and NLU services. And we are hiring!
Plan

- **Weak Supervision NLP**
  - NLP, AI, software 2.0
  - Semantics as a foreign language
  - Unsupervised learning
  - Knowledge representation (symbolism)

- **Semantic Parsing Tasks**
  - WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines** (ACL 2017)
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)
  - RL vs MLE

- **Memory Augmented Policy Optimization** (NIPS 2018)
  - Experience replay (long term memory & optimal updating strategy)
  - Systematic exploration
  - Memory Weight Clipping (unbiased cold start strategy)

Access slides and join discussions at weakly-supervised-nlu google group
Natural Language Processing (NLP)

- Enables machines to understand and assists human
- A major problem of AI (AI-complete)
- Leads to new theories in cognitive science

There can be two underlying motivations for building a computational theory. The **technological goal** is simply to build better computers, and any solution that works would be acceptable. The **cognitive goal** is to build a computational analog of the human-language-processing mechanism; such a theory would be acceptable only after it had been verified by experiment. -- James Allen, 1987
Adoption Of Voice Technology

- Google’s Speech Internationalization Project: From 1 to 300 Languages and Beyond [Pedro J. Moreno, 2012]
- My daughter adopted YouTube voice command since 2 years old
- 20% of the U.S. population has access to smart speakers [Techcrunch, 2018]
- Rising adoption in the Asia Pacific
If you got a billion dollars to spend on a huge research project that you get to lead, what would you like to do?
-- r/CyberByte, 2015

NLP is fascinating, allowing us to focus on highly-structured inference problems, on issues that go to the core of "what is thought" but remain eminently practical, and on a technology that surely would make the world a better place.
-- Michael I Jordan

What kind of impact you hope deep learning has on our future?
-- Steve Paikin, 2016

I hope it allows Google to ... search by the content of the document rather than by the words in the document ... I hope it will make for intelligent personal systems, who can answer questions in a sensible way ... It will make computers much easier to use. Because you'll be able to just say to your computer "print this damn thing"
-- Geoffrey Hinton
What is understanding?

“If they find a parrot who could answer to everything, I would claim it to be an intelligent being without hesitation.”
-- Alan Turing, 1950

Does the machine literally “understand” Chinese? Or is it merely simulating the ability to understand Chinese?
-- John Searle, 1980

The Imitation Game

The Chinese Room Argument
Full Supervision NLP

- Traditionally NLP is a labor-intensive business

Applications

<table>
<thead>
<tr>
<th>Discourse Processing</th>
<th>Semantic Parsing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic analysis</td>
<td>Morphological analysis</td>
</tr>
</tbody>
</table>

ASR | OCR

text | speech

NLP algorithms simply lookup

Rules written by 1000 PhDs

Rules written by 1000 PhDs

NLP algorithms simply lookup
Software 2.0

1. specify some goal on the behavior
   ○ e.g., “satisfy input output pairs of examples”,
   ○ e.g., “win a game of Go”

2. write a rough skeleton of the code that identifies a subset of program space to search
   ○ e.g. a neural net architecture

3. use the computational resources at disposal to search this space for a program that works.

Death of feature engineering. (The) users of the software will (play) a direct role in building it. Data labeling is a central component to system design.
Where does knowledge come from?

- Can only cover the most popular semantics used by human

the world → domain experts (bottlenecks) → expert systems with knowledge bases
Weak Supervision NLP

- Avoid the knowledge acquisition bottleneck with machine learning
- Then we can cover all possible semantics used by human intelligent systems with knowledge

end to end examples (e.g., QA pairs)  \[\rightarrow\]  machine learning  \[\rightarrow\]  intelligent systems with knowledge
Language & Reasoning

- The formalist view
  - Language was primarily invented for reasoning [Everaert+ 2015]

- The functionalist view
  - Language is for communication [Kirby 2017]

- Cognitive coupling hypothesis
  - Sequential processing is “necessary for behaviours such as primate tool use, navigation, foraging and social action.” [Kolodny & Edelman 2018]
Reasoning is needed to understand text

[Bart's father is Homer]

[Liag+ 2017]
Semantics is a language for computation

“impressionist painters during the 1920s”

impressionist <visual_artist> x.[/associated_periods_or_movements = /impressionism]

<artist> during the 1920s x.[/date_of_work < 1930; /date_of_work > 1920]

painters [/painting] !/art_forms
Largest city in US?

Semantics is a language for computation

GO
(Hop V1 CityIn)
(Argmax V2 Population)
RETURN

NYC

Freebase, DBpedia, YAGO, NELL
Semantics as a foreign language

1) **Natural languages** are programming languages to **control** human behavior (either others or self)

2) For machines and human to understand each other, they just need **translation** models trained with **control theory**
NLU with Unsupervised Learning

- **Supervised** learning needs either feature engineering for compact representation or large amount of labeled data.

- **Unsupervised** learning produces better representations and reduces the labeling cost, and it is easily transferable!
“Pure” Reinforcement Learning (cherry)
- The machine predicts a scalar reward given once in a while.
- A few bits for some samples

Supervised Learning (icing)
- The machine predicts a category or a few numbers for each input
- Predicting human-supplied data
- 10→10,000 bits per sample

Unsupervised/Predictive Learning (cake)
- The machine predicts any part of its input for any observed part.
- Predicts future frames in videos
- Millions of bits per sample
Pre-Trained Word Embeddings

- Learning **non-contextual** word embeddings from text
  - matrix factorization on global **word-word co-occurrence counts** [1990]
  - local context window methods [2014] will come back to this in 2018
  - weighted least squares on global **word-word co-occurrence counts** [2014]

- Co-occurrence statistics as an efficient approximation to the original text

<table>
<thead>
<tr>
<th>Probability and Ratio</th>
<th>$k = \text{solid}$</th>
<th>$k = \text{gas}$</th>
<th>$k = \text{water}$</th>
<th>$k = \text{fashion}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(k</td>
<td>\text{ice})$</td>
<td>$1.9 \times 10^{-4}$</td>
<td>$6.6 \times 10^{-5}$</td>
<td>$3.0 \times 10^{-3}$</td>
</tr>
<tr>
<td>$P(k</td>
<td>\text{steam})$</td>
<td>$2.2 \times 10^{-5}$</td>
<td>$7.8 \times 10^{-4}$</td>
<td>$2.2 \times 10^{-3}$</td>
</tr>
<tr>
<td>$P(k</td>
<td>\text{ice}) / P(k</td>
<td>\text{steam})$</td>
<td>$8.9$</td>
<td>$8.5 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

$$J = \sum_{i,j=1}^{V} f \left( X_{ij} \right) \left( w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$
Pre-Trained Word Embeddings

Compact representations for **syntax, common sense** and **world knowledge**

- LSA [Deer-wester+ 1990]
- skip-gram [Mikolov + 2013]
- Glove [Pennington+ 2014]
Pre-Trained Sequence Models

With a lot more computational power we have language modeling sequence models which converts sequences of tokens to sequences of contextual embeddings.

(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER
Pre-Trained Sequence Models

Differences in pre-training model architectures.
BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM.

[Devlin+ 2018]
[Radford+ 2018]
[Peters+ 2018]
Pre-Trained Sequence Models

2018 is the year of Pre-Trained Sequence Models

<table>
<thead>
<tr>
<th>System</th>
<th>MNLI-(m/mm)</th>
<th>QQP</th>
<th>QNLI</th>
<th>SST-2</th>
<th>CoLA</th>
<th>STS-B</th>
<th>MRPC</th>
<th>RTE</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>392k</td>
<td>363k</td>
<td>108k</td>
<td>67k</td>
<td>8.5k</td>
<td>5.7k</td>
<td>3.5k</td>
<td>2.5k</td>
<td></td>
</tr>
<tr>
<td>Pre-OpenAI SOTA</td>
<td>80.6/80.1</td>
<td>66.1</td>
<td>82.3</td>
<td>93.2</td>
<td>35.0</td>
<td>81.0</td>
<td>86.0</td>
<td>61.7</td>
<td>74.0</td>
</tr>
<tr>
<td>BiLSTM+ELMo+Attn</td>
<td>76.4/76.1</td>
<td>64.8</td>
<td>79.9</td>
<td>90.4</td>
<td>36.0</td>
<td>73.3</td>
<td>84.9</td>
<td>56.8</td>
<td>71.0</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>82.1/81.4</td>
<td>70.3</td>
<td>88.1</td>
<td>91.3</td>
<td>45.4</td>
<td>80.0</td>
<td>82.3</td>
<td>56.0</td>
<td>75.2</td>
</tr>
<tr>
<td>BERT_{BASE}</td>
<td>84.6/83.4</td>
<td>71.2</td>
<td>90.1</td>
<td>93.5</td>
<td>52.1</td>
<td>85.8</td>
<td>88.9</td>
<td>66.4</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT_{LARGE}</td>
<td><strong>86.7/85.9</strong></td>
<td><strong>72.1</strong></td>
<td><strong>91.1</strong></td>
<td><strong>94.9</strong></td>
<td><strong>60.5</strong></td>
<td><strong>86.5</strong></td>
<td><strong>89.3</strong></td>
<td><strong>70.1</strong></td>
<td><strong>81.9</strong></td>
</tr>
</tbody>
</table>
Knowledge Representation & Scalability

a small machine which copies the complexity of the world to the brain

the world

a suitable representation
Mandelbrot Set

the nature of complex numbers

\[ z_0 = 0 \]

\[ z_{n+1} = z_n^2 + c \]

\[ c \in M \iff \lim_{n \to \infty} |z_{n+1}| \leq 2 \]
Internet as an external memory

- How information should be organized for scalability?

Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and to coin one at random, memex will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.
The scalability of modern search engines

- Can respond to user's requests within a fraction of a second
- But are weak at text understanding and complex reasoning

[Brin & Page 1998]
Can we search entities on the web?

- Multimedia, business, products have a lot of reviews and descriptions
Traditional IR approach lacks understanding

- Need to interpret the meaning from the surface text
Question answering as a simple test bed

- A good semantic representation should support reasoning at scale
Three approaches to generative models

- Autoregression (e.g., LM), VAE, GAN

Autoregressive models (e.g. LM)
[Hochreiter & Schmidhuber 1997]
Graves [1308.0850]

Variational Autoencoders (VAE)
Kingma and Welling
[1312.6114]

Generative Adversarial Networks (GAN)
Goodfellow et al. [1406.2661]

Blog [Karpathy+ 2016]
Immediately criticised when applied to text

- "I have a lot of respect for language. Deep-learning people seem not to"
- "They include such impressive natural language sentences as:
  - * what everything they take everything away from
  - * how is the antoher headache
  - * will you have two moment ?
  - * This is undergoing operation a year .
- "These are not even grammatical!"
- The DNN bubble consists of models, which show great promises but not yet practical at this point
statistician’s view v.s. linguist’s view

Given the power of deep learning anything can be mapped to a unit Gaussian ball

The world has real structures, which need to be represented by real structures

furious

Seq2Seq [Sutskever, Vinyals, Le 2014]
VAE [Kingma & Welling 2014]
GAN [Goodfellow+ 2014]
ACL [Goldberg 2015]
ACL [Mooney 2015]
Scalability of mammal memory

- Very rapid adaptation (in just one or a few trials) is necessary for survival
  - E.g., associating taste of food and sickness

- Need fast responses based on large amount of knowledge
  - Needs good representation of knowledge

- However, good representation can only be learnt gradually
  - During sleeps to prevent interference with established associations

Scalability of Cognitive Architectures

- The design of mammalian brains is inspiring to NLP systems
  - they are solving similar problems

- The design has not changed much since 30 years ago
  - “We’ve totally solved it already ... it is just a matter of job security”
    -- Nate Derbinsky, Northeastern U.

- Today we have
  - internet economy and data
  - computation and ML development
Plan

● **Weak Supervision NLP**
  ○ NLP, AI, software 2.0
  ○ Semantics as a foreign language
  ○ Unsupervised learning
  ○ Knowledge representation (symbolism)

● **Semantic Parsing Tasks**
  ○ WebQuestionsSP, WikiTableQuestions

● **Neural Symbolic Machines** (ACL 2017)
  ○ Compositionality (short term memory)
  ○ Scalable KB inference (symbolism)
  ○ RL vs MLE

● **Memory Augmented Policy Optimization** (NIPS 2018)
  ○ Experience replay (long term memory & optimal updating strategy)
  ○ Systematic exploration
  ○ Memory Weight Clipping (unbiased cold start strategy)

Access slides and join discussions at weakly-supervised-nlu google group
Semantic Parsing

- Natural language queries or commands are converted to computation steps on data and produce the expected answers or behavior.

[Berant+ 2013] [Liang 2013]
Related Works

- Training from full supervision is labor-intensive
  - DeepCoder [Balog, 2016]
  - NPI [Reed & Freitas, 2015]
  - Seq2Tree [Dong & Lapata, 2016]
  - ...

- Traditional semantic parsing models require feature engineering
  - SEMPRE [Berant et al, 2013]
  - STAGG [Yih et al, 2015]
  - ...

- End-to-end differentiable models cannot scale to large databases
  - Neural Programmer [Neekalantan et al, 2015]
  - Neural Turing Machines [Graves et al, 2014]
  - Neural GPU [Kaiser & Sutskever, 2015]
  - ...

- Combine deep learning, symbolic reasoning and reinforcement learning
  - Neural Symbolic Machines (Liang, Berant, Le, Forbus, Lao, 2017)
  - Memory Augmented Policy Optimization (MAPO) (Liang, Norouzi, Berant, Le, Lao, 2018)
Question Answering with Knowledge Base

Largest city in US?

GO
(Hop V1 CityIn)
(Argmax V2 Population)
RETURN

NYC

Paraphrase
Many ways to ask the same question, e.g.,
“What was the date that Minnesota became a state?”
“When was the state Minnesota created?”

Compositionality

Paraphrase

Large Search Space
(Optimization)
E.g., Freebase:
23K predicates,
82M entities,
417M triplets
WebQuestions: motivation

**Motivation:** Natural language interface to large structured knowledge-bases

**Background:** availability of many structured datasets (Google KG, Bing Satori, Freebase, DBPedia, Yelp, ...)

Introducing the Knowledge Graph: things, not strings
WebQuestions: getting questions

Goal: collect large number of natural language queries

Strategy: breadth-first search over Google Suggest graph results in 1M queries

Where was Barack Obama born?

Where was ___ born?

Google Suggest
Barack Obama
Lady Gaga
Steve Jobs

Where was Steve Jobs born?

Where was Steve Jobs ___?

Google Suggest
born raised
on the Forbes list

Where was Steve Jobs raised?
Motivation: overcome the limitations of computers -- machines struggle to absorb knowledge in the way humans do.

Solution: a large collaborative knowledge base consisting of data (int triple format) composed mainly by its community members.

Later acquired by Google (discontinued)

41M entities (nodes)
19K properties (edge labels)
596M assertions (edges)
WebQuestions: getting Freebase answers [Berant+ 2013]

**Goal**: obtain label from non-experts

**Strategy**: Amazon Mechanical Turk (AMT)

- Given a query e.g. “Eric Clapton hometown” detect if there is a single named entity “Eric Clapton”
- Ask the worker to pick one (or more) of the possible entities/values (“Ripley”) on the entity Freebase page
- Cost $0.03 per question
WebQuestionsSP Dataset

- 5,810 questions from Google Suggest API & Amazon MTurk
- Remove invalid QA pairs
- 3,098 training examples, 1,639 testing examples remaining
- Open-domain and contains grammatical error
- Multiple entities as answer => macro-averaged F1

[Berant et al, 2013; Yih et al, 2016]

- What **do** Michelle Obama do for a living?
- What character did Natalie Portman play in Star Wars?
- What currency do you use in Costa Rica?
- What did Obama study in school?
- What killed Sammy Davis Jr?

Grammatical error

Multiple entities

writer, lawyer

Padme Amidala

Costa Rican colon

political science

throat cancer
SEMPRE: paraphrase

Collect entity pair observations for phrases and augmented predicates (which partially solves the search problem)

ClueWeb09
1 billion docs

15M triples (of 15k phrases)
(Barack Obama, was born in, Honolulu)
(Albert Einstein, was born in, Ulm)
(Barack Obama, lived in, Chicago)
... ...

ReVerb

600M triples (of 60k augmented predicates)
(BarackObama, PlaceOfBirth, Honolulu)
(Albert Einstein, PlaceOfBirth, Ulm)
(BarackObama, PlacesLived.Location, Chicago)
... ...

[Berant+ 2013]
[Lin+ 2012]
[Fader+ 2011]
SEMPRE: paraphrase

- Construct lexicon and alignment features based on entity pair cooccurrences

\[\text{grew up in}[\text{Person}, \text{Location}] \rightarrow \text{DateOfBirth}\]
\[\text{born in}[\text{Person}, \text{Date}] \rightarrow \text{PlaceOfBirth}\]
\[\text{married in}[\text{Person}, \text{Date}] \rightarrow \text{Marriage.StartDate}\]
\[\text{born in}[\text{Person}, \text{Location}] \rightarrow \text{PlacesLived.Location}\]

**Alignment features**
- log-phrase-count: \(\log(15765)\)
- log-predicate-count: \(\log(9182)\)
- log-intersection-count: \(\log(6048)\)
- KB-best-match: 0
SEMPRE: compositionality

The **semantic** structure (**logic form**) is coupled with **syntactic** structure (**surface patterns**) through **composition rules**

**One derivation**

\[
\text{Type.Location} \cap \text{PeopleBornHere.BarackObama} \\
\text{intersection} \\
\text{Type.Location} \quad \text{was} \quad \text{PeopleBornHere.BarackObama} \quad ? \\
\text{lexicon} \\
\text{where} \\
\text{BarackObama} \quad \text{PeopleBornHere} \\
\text{lexicon} \\
\text{Obama} \quad \text{born}
\]
**Motivation:** individual words can be highly ambiguous
- What government does Chile have?
- What is Italy's language?
- Where is Beijing?
- What is the cover price of X-men?

**Solution:** the **bridging** operation generates predicates compatible with neighboring predicates.

Occidental College, Columbia University

Execute on Database

Type.University $\sqcap$ Education.BarackObama

Type.University

bridging

Education

BarackObama

alignment

Which
college
did
Obama
go to
SEMPRE: modeling

Log linear model over derivations $d$ given utterance $x$ with expert designed features

$$p_{\theta}(d \mid x) = \frac{\exp\{\phi(x,d) \top \theta\}}{\sum_{d' \in D(x)} \exp\{\phi(x,d') \top \theta\}}$$

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>Log of # entity pairs that occur with the phrase $r_1 (</td>
</tr>
<tr>
<td></td>
<td>Log of # entity pairs that occur with the logical predicate $r_2 (</td>
</tr>
<tr>
<td></td>
<td>Log of # entity pairs that occur with both $r_1$ and $r_2 (</td>
</tr>
<tr>
<td></td>
<td>Whether $r_2$ is the best match for $r_1 (r_2 = \arg \max_r</td>
</tr>
<tr>
<td>Lexicalized</td>
<td>Conjunction of phrase $w$ and predicate $z$</td>
</tr>
<tr>
<td>Text similarity</td>
<td>Phrase $r_1$ is equal/prefix/suffix of $s_2$</td>
</tr>
<tr>
<td></td>
<td>Phrase overlap of $r_1$ and $s_2$</td>
</tr>
<tr>
<td>Bridging</td>
<td>Log of # entity pairs that occur with bridging predicate $b (</td>
</tr>
<tr>
<td></td>
<td>Kind of bridging (# unaries involved)</td>
</tr>
<tr>
<td></td>
<td>The binary $b$ injected</td>
</tr>
<tr>
<td>Composition</td>
<td># of intersect/join/bridging operations</td>
</tr>
<tr>
<td></td>
<td>POS tags in join/bridging and skipped words</td>
</tr>
<tr>
<td></td>
<td>Size of denotation of logical form</td>
</tr>
</tbody>
</table>

Table 1: Full set of features. For the alignment and text similarity, $r_1$ is a phrase, $r_2$ is a predicate with Freebase name $s_2$, and $b$ is a binary predicate with type signature $(t_1,t_2)$. 49
STAGG: paraphrase

3 matching models based on char-ngram conv nets (Sent2vec [Shen+ 14])

Pattern-Chain: who voiced meg on <e>  cast-actor
Question-EntPred: who voiced meg on family guy  Meg Griffin cast-actor
ClueWeb12: voiced homer on <e>  cast-actor
STAGG: search

Staged Query Graph Generation

“Who first voiced Meg on Family Guy?”

1) decide the topic entity

2) decide the core inference chain

3) add type/aggregation constraints
STAGG: search

keeps up to N candidate states in the priority queue (N = 1000)

+10 special rules to restrict the type/aggregation constraints

e.g. Consider “to” predicates (indicating the ending time of an event) only when the question contains “last”, “latest” or “newest”

Domain knowledge is often very important for structure search problems. Will revisit in WikiTableQ

---

Algorithm 1 Staged query graph generation

**Require:** Priority queue $H$ with limited size $N$

1: $s_o \leftarrow \phi$; $r_o \leftarrow -\infty$
2: $H$.add($s_o$, $r_o$)
3: while $H$ is not empty do
4:   $s$, $r$ $\leftarrow$ $H$.pop()
5:   if $r > r_o$ then
6:     $s_o \leftarrow s$; $r_o \leftarrow r$
7:   end if
8: for all $a \in \Pi(s)$ do
9:   $s' \leftarrow T(s, a)$
10:  $H$.add($s'$, $\gamma(s')$)
11: end for
12: end while
13: return $s_o$
STAGG: modeling

Learning to rank model (for query graph candidates) based on expert designed features + 10 special features on the type/aggregation constraints

$q = “Who first voiced Meg on Family Guy?”$

(1) EntityLinkingScore(FamilyGuy, “Family Guy”) = 0.9
(2) PatChain(“who first voiced meg on <e>”, cast-actor) = 0.7
(3) QuesEP(q, “family guy cast-actor”) = 0.6
(4) ClueWeb(“who first voiced meg on <e>”, cast-actor) = 0.2
(5) ConstraintEntityWord(“Meg Griffin”, q) = 0.5
(6) ConstraintEntityInQ(“Meg Griffin”, q) = 1
(7) AggregationKeyword(argmin, q) = 1
(8) NumNodes(s) = 5
(9) NumAns(s) = 1
Brief Summary & Preview

DL leads to the death of feature engineering (but not domain knowledge)
DL makes it easier to leverage pre-trained unsupervised models

<table>
<thead>
<tr>
<th></th>
<th>SEMPRE/STAGG</th>
<th>This talk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paraphrase</td>
<td>Co-occurrences collected from 1B docs (ClueWeb) and Freebase</td>
<td>Embeddings trained from 840B text tokens (GloVe)</td>
</tr>
<tr>
<td>(semi-supervised)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compositionality</td>
<td>Relies on syntactic structure (text spans) and domain specific rules to constrain the generation of logic forms</td>
<td>A (LISP like) language specify computations on KG</td>
</tr>
<tr>
<td>Search</td>
<td>priority queue &amp; heuristic rules</td>
<td>RL &amp; a program interpreter with syntactical &amp; semantic checks</td>
</tr>
<tr>
<td>Modeling</td>
<td>Expert designed features</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>
WikiTableQuestions: Dataset

**Breadth**
- No fixed schema: Tables at test time are not seen during training, needs to generalize based on column name.

**Depth**
- More compositional questions, thus require longer programs
- More operations like arithmetic operations and aggregation operations

### Example Questions

- $x_1$: “Greece held its last Summer Olympics in which year?”
  $y_1$: \{2004\}

- $x_2$: “In which city’s the first time with at least 20 nations?”
  $y_2$: \{Paris\}

- $x_3$: “Which years have the most participating countries?”
  $y_3$: \{2008, 2012\}

- $x_4$: “How many events were in Athens, Greece?”
  $y_4$: \{2\}

- $x_5$: “How many more participants were there in 1900 than in the first year?”
  $y_5$: \{10\}

<table>
<thead>
<tr>
<th>Year</th>
<th>City</th>
<th>Country</th>
<th>Nations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1896</td>
<td>Athens</td>
<td>Greece</td>
<td>14</td>
</tr>
<tr>
<td>1900</td>
<td>Paris</td>
<td>France</td>
<td>24</td>
</tr>
<tr>
<td>1904</td>
<td>St. Louis</td>
<td>USA</td>
<td>12</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2004</td>
<td>Athens</td>
<td>Greece</td>
<td>201</td>
</tr>
<tr>
<td>2008</td>
<td>Beijing</td>
<td>China</td>
<td>204</td>
</tr>
<tr>
<td>2012</td>
<td>London</td>
<td>UK</td>
<td>204</td>
</tr>
</tbody>
</table>

[Pasupat & Liang, 2015]
# WikiTableQuestions: semantics

<table>
<thead>
<tr>
<th>Function</th>
<th>Arguments</th>
<th>Returns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{hop} v p )</td>
<td>( v ): a list of rows. ( p ): a column.</td>
<td>a list of cells.</td>
<td>Select the given column of the given rows.</td>
</tr>
<tr>
<td>( \text{argmax} v p ) ( \text{argmin} v p )</td>
<td>( v ): a list of rows. ( p ): a number or date column.</td>
<td>a list of rows.</td>
<td>From the given rows, select the ones with the largest / smallest value in the given column.</td>
</tr>
<tr>
<td>( \text{filter}<em>&gt; v q p ) ( \text{filter}</em>\geq v q p ) ( \text{filter}<em>&lt; v q p ) ( \text{filter}</em>\leq v q p ) ( \text{filter}<em>= v q p ) ( \text{filter}</em>\neq v q p )</td>
<td>( v ): a list of rows. ( q ): a number or date. ( p ): a number or date column.</td>
<td>a list of rows.</td>
<td>From the given rows, select the ones whose given column has certain order relation with the given value.</td>
</tr>
<tr>
<td>( \text{filter}<em>{in} v q p ) ( \text{filter}</em>{\not= in} v q p )</td>
<td>( v ): a list of rows. ( q ): a string. ( p ): a string column.</td>
<td>a list of rows.</td>
<td>From the given rows, select the ones whose given column contain / do not contain the given string.</td>
</tr>
</tbody>
</table>
## WikiTableQuestions: semantics

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(first v)</strong></td>
<td></td>
<td>Select the row with the smallest index from the given rows.</td>
</tr>
<tr>
<td><strong>(last v)</strong></td>
<td></td>
<td>Select the row with the largest index from the given rows.</td>
</tr>
<tr>
<td><strong>(previous v)</strong></td>
<td>v: a row.</td>
<td>Select the row above the given row.</td>
</tr>
<tr>
<td><strong>(next v)</strong></td>
<td>v: a row.</td>
<td>Select the row below the given row.</td>
</tr>
<tr>
<td><strong>(count v)</strong></td>
<td>v: a list of rows.</td>
<td>Count the number of rows.</td>
</tr>
<tr>
<td><strong>(max v p)</strong></td>
<td>v: a list of rows.</td>
<td>p: a number column. Compute the maximum value in the given column.</td>
</tr>
<tr>
<td><strong>(min v p)</strong></td>
<td>v: a list of rows.</td>
<td>p: a number column. Compute the minimum value in the given column.</td>
</tr>
<tr>
<td><strong>(average v p)</strong></td>
<td>v: a list of rows.</td>
<td>p: a number column. Compute the average value in the given column.</td>
</tr>
<tr>
<td><strong>(sum v p)</strong></td>
<td>v: a list of rows.</td>
<td>p: a number column. Compute the sum value in the given column.</td>
</tr>
<tr>
<td><strong>(mode v p)</strong></td>
<td>v: a list of rows.</td>
<td>p: a column. Get the most common value in the given column.</td>
</tr>
<tr>
<td><strong>(same_as v p)</strong></td>
<td>v: a row.</td>
<td>p: a column. Get the rows where the column is the same as the given row.</td>
</tr>
<tr>
<td><strong>(diff v0 v1 p)</strong></td>
<td>v0: a row.</td>
<td>v1: a row. p: a number column. Compute the difference in the given column.</td>
</tr>
</tbody>
</table>

* *same_as* function requires the given column to have the same data type as the cells in the rows.
Feature engineering is dead. It is survived by program space design.

(when t-alternative
  (rule $AnchoredOr ($LEMMA_TOKEN) (FilterTokenFn lemma and or) (anchored 1))
...)

(when t-movement
  (rule $AnchoredMovement ($LEMMA_TOKEN) (FilterTokenFn lemma next previous after before above below) (anchored 1))
...)

(when t-comparison
  (rule $AnchoredMore ($LEMMA_TOKEN) (FilterTokenFn lemma more than least above after) (anchored 1))
...)

(when t-superlative
  (rule $SuperlativeTrigger ($LEMMA_TOKEN) (FilterPosTagFn token JJR JJS RBR RBS) (anchored 1))
  (rule $SuperlativeTrigger ($LEMMA_TOKEN) (FilterTokenFn lemma top first bottom last) (anchored 1))
...)

(when t-arithmetic
  (rule $AnchoredSub ($LEMMA_TOKEN) (FilterTokenFn lemma difference between and much) (anchored 1))
...)

[Pasupat & Liang, 2015]
[Zhang, Pasupat & Liang, 2017]
[Liang+ 2018]
## WikiTableQuestions: example solutions

### Superlative

<table>
<thead>
<tr>
<th>Question</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nt-13901: the most points were scored by which player?</strong></td>
<td>(argmax all_rows r.points-num) Sort all rows by column ‘points’ and take the first row.</td>
</tr>
<tr>
<td>(hop v0 r.player-str)</td>
<td>Output the value of column ‘player’ for the rows in v0.</td>
</tr>
</tbody>
</table>

### Difference

<table>
<thead>
<tr>
<th>Question</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>nt-457: how many more passengers flew to los angeles than to saskatoon?</strong></td>
<td>(filter(_{in}) all_rows [’saskatoon’] r.city-str) Find the row with ‘saskatoon’ matched in column ‘city’.</td>
</tr>
<tr>
<td>(filter(_{in}) all_rows [’los angeles’] r.city-str)</td>
<td>Find the row with ‘los angeles’ matched in column ‘city’.</td>
</tr>
<tr>
<td>(diff v1 v0 r.passengers-num)</td>
<td>Calculate the difference of the values in the column ‘passenger’ of v0 and v1.</td>
</tr>
</tbody>
</table>
More examples

<table>
<thead>
<tr>
<th>Before / After</th>
<th>nt-10832: which nation is before peru?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(filter$_{in}$ all_rows ['peru'] r.nation-str)</td>
</tr>
<tr>
<td></td>
<td>(previous v0)</td>
</tr>
<tr>
<td></td>
<td>(hop v1 r.nation-str)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Compare &amp; Count</th>
<th>nt-647: in how many games did sri lanka score at least 2 goals?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(filter$_{\geq}$ all_rows [2] r.score-num)</td>
</tr>
<tr>
<td></td>
<td>(count v0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exclusion</th>
<th>nt-1133: other than william stuart price, which other businessman was born in tulsa?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(filter$_{in}$ all_rows ['tulsa'] r.hometown-str)</td>
</tr>
<tr>
<td></td>
<td>(filter!$_{in}$ v0 ['william stuart price'] r.name-str)</td>
</tr>
<tr>
<td></td>
<td>(hop v1 r.name-str)</td>
</tr>
</tbody>
</table>
Neural Program Induction & Scalabilities

- Impressive works to show NN can learn addition and sorting, but...

  ![Addition scratch pad](image1)

  ![Input array](image2)

- The learned operations are not as scalable and precise.

  ![NPI inference](image3)

  ![Generated commands](image4)

  ![Sorting per-sequence accuracy vs sequence length](image5)

- Why not use existing modules that are scalable, precise and interpretable?

  ![Google search](image6)

  [Reed & Freitas 2015]

  [Zaremba & Sutskever 2016]
Plan

- **Weak Supervision NLP**
  - NLP, AI, software 2.0
  - Semantics as a foreign language
  - Unsupervised learning
  - Knowledge representation (symbolism)

- **Semantic Parsing Tasks**
  - WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines** (ACL 2017)
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)
  - RL vs MLE

- **Memory Augmented Policy Optimization** (NIPS 2018)
  - Experience replay (long term memory & optimal updating strategy)
  - Systematic exploration
  - Memory Weight Clipping (unbiased cold start strategy)

Access slides and join discussions at weakly-supervised-nlu google group
A bit background on Seq2Seq

- Separate a sequence model into encoder and decoder
- Improves a phrase-based SMT system by re-ranking top candidates
- Cannot perform well by itself due to the information bottleneck

[Hochreiter & Schmidhuber 1997]
[Sutskever, Vinyals, Le 2014]
Re-thinking sequence-to-sequence learning

- Cooperation among three agents

1. **Agent 1** (Encoder): transforms the source sentence into a set of code vectors in a memory

2. **Agent 2** (Search): searches for relevant code vectors in the memory based on the command from the Agent 3 and returns them to the Agent 3.

3. **Agent 3** (Decoder): observes the current state (previously decoded symbols), commands the Agent 2 to find relevant code vectors and generates the next symbol based on them.
Re-thinking sequence-to-sequence learning

1. Don't generate from the ball of $Z$

2. Generate from a sequence of source token ids, which encodes the semantics of the target sentence
Language to Logical Form with Neural Attention

- "compared to previous methods our model achieves similar or better performance .. with no hand-engineered .. features."
- Relies on full supervision (labeled logical forms)
LSTM & Lapata’s scream

- LSTM has been applied to all kinds of NLP tasks and has greatly simplified system designs
What now? Is NLP DEAD?

- The need for **symbolic operations** and **effective model optimization**

<table>
<thead>
<tr>
<th></th>
<th>SEMPRE/STAGG</th>
<th>This talk</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Paraphrase (semi-supervised)</strong></td>
<td>Co-occurrences collected from 1B docs (ClueWeb) and Freebase</td>
<td>Embeddings trained from 840B text tokens (GloVe)</td>
</tr>
<tr>
<td><strong>Compositionality</strong></td>
<td>Relies on syntactic structure (text spans) and domain specific rules to constrain the generation of logic forms</td>
<td>A (LISP like) language specify computations on KG</td>
</tr>
<tr>
<td><strong>Search</strong></td>
<td>priority queue &amp; heuristic rules</td>
<td>RL &amp; a program interpreter with syntactical &amp; semantic checks</td>
</tr>
<tr>
<td><strong>Modeling</strong></td>
<td>Expert designed features</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>
Connectionism vs Symbolism

The symbolic models represent elegant solutions to problems, and have been dominating AI for a very long time.

Once we have figured out how to train them, the connectionism approaches start to win.

VS.

VS.

a symbolic machine

a neural controller

a sequence of symbols
Symbolic Machines in Brains

- 2014 Nobel Prize in Physiology or Medicine awarded for ‘inner GPS’ research
- Positions are represented as discrete representations in animals' brains, which enable accurate and autonomous calculations

![Diagram showing relationships between Environment, Brain, and Symbolic Modules](Stensola+ 2012)

- Location cells & grid cells
- Grid spacing for all modules (M1–M4) in different animals

![Graph showing grid spacing](Stensola+ 2012)
Neural Symbolic Machines

Weak supervision
Manager

Neural
Programmer

Symbolic
Computer

Question
Answer

Program
Results

Knowledge Base
Predefined Functions

Abstract
Scalable
Precise
Non-differentiable

HERE’S ANOTHER SHOVEL FULL OF ASSIGNMENTS.
Computer with Domain Specific Languages

- Lisp Interpreter
  - Program => \( \text{exp}_1 \text{exp}_2 \ldots \text{exp}_n \) <END>
  - \( \text{Exp} \Rightarrow (\text{f arg}_1 \text{arg}_2 \ldots \text{arg}_n) \)

- What functions will be useful for the given task?
  - 10 operations for WebQuestions
  - 22 different operations for WikiTableQuestions
    - hop
    - argmax, argmin
    - filter\(_=\), filter\(_!=\), filter\(_>\), filter\(_<\), filter\(_>=\), filter\(_<=\), filter\(_\in\), filter\(_\notin\)
    - first, last, previous, next
    - max, min, average, sum, mode, diff, same

ACL [Liang+ 2017]
Code Assistance to Prune Search Space

ACL [Liang+ 2017]
# Code Assistance: Syntactic Constraint

## Decoder Vocab

<table>
<thead>
<tr>
<th>$V_0$</th>
<th>R0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>R1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$E_0$</td>
<td>Hop</td>
</tr>
<tr>
<td>$E_1$</td>
<td>Argmax</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$P_0$</td>
<td>CityIn</td>
</tr>
<tr>
<td>$P_1$</td>
<td>BornIn</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Variables:** <10
- **Functions:** <10
- **Predicates:** 23K

Last token is ‘(’, so has to output a function name next.
Code Assistance: Semantic Constraint

Decoder Vocab

<table>
<thead>
<tr>
<th>V₀</th>
<th>R₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>V₁</td>
<td>R₁</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>E₀</td>
<td>Hop</td>
</tr>
<tr>
<td>E₁</td>
<td>Argmax</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>P₀</td>
<td>CityIn</td>
</tr>
<tr>
<td>P₁</td>
<td>BornIn</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- Variables: <10
- Functions: <10
- Predicates: 23K
Given definition of $\text{Hop}$, need to output a predicate that is connected to $R_2$ (m.USA).

Decoder Vocab

- Variables: <10
- Functions: <10
- Predicates: 23K
- Valid Predicates: <100
• Equivalent to a linearised bottom-up derivation of the recursive program
Save Intermediate Values

<table>
<thead>
<tr>
<th>Key (Embedding)</th>
<th>Variable (Symbol)</th>
<th>Value (Data in Computer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_0$</td>
<td>R0</td>
<td>m.USA</td>
</tr>
<tr>
<td>$V_1$</td>
<td>R1</td>
<td>[m.SF, m.NYC, ...]</td>
</tr>
</tbody>
</table>

Expression is finished.

ACL [Liang+ 2017]
## Reuse Intermediate Values

### Table

<table>
<thead>
<tr>
<th>Key (Embedding)</th>
<th>Variable (Symbol)</th>
<th>Value (Data in Computer)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_0$</td>
<td>R0</td>
<td>m.USA</td>
</tr>
<tr>
<td>$V_1$</td>
<td>R1</td>
<td>[m.SF, m.NYC, ...]</td>
</tr>
</tbody>
</table>

### Diagram

- **Input:** !CityIn
- **Softmax**
- **Argmax**
- **Output:** (m.USA, [m.SF, m.NYC, ...])

### Reference

ACL [Liang+ 2017]
Augmented REINFORCE

- REINFORCE get stuck at local maxima
- Iterative ML training is not directly optimizing the F1 score
- Augmented REINFORCE obtains better performances

<table>
<thead>
<tr>
<th>Settings</th>
<th>Train Avg. F1@1</th>
<th>Valid Avg. F1@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>iterative ML only</td>
<td>68.6</td>
<td>60.1</td>
</tr>
<tr>
<td>REINFORCE only</td>
<td>55.1</td>
<td>47.8</td>
</tr>
<tr>
<td>Augmented REINFORCE</td>
<td>83.0</td>
<td><strong>67.2</strong></td>
</tr>
</tbody>
</table>
State-of-the-Art on WebQuestionsSP

- First end-to-end neural network to achieve SOTA on semantic parsing with weak supervision over large knowledge base
- The performance is approaching SOTA with full supervision

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg. Prec.@1</th>
<th>Avg. Rec.@1</th>
<th>Avg. F1@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{STAGG}</td>
<td>67.3</td>
<td>73.1</td>
<td>66.8</td>
</tr>
<tr>
<td>\textit{NSM – our model}</td>
<td>70.8</td>
<td>76.0</td>
<td>69.0</td>
</tr>
<tr>
<td>\textit{STAGG (full supervision)}</td>
<td>70.9</td>
<td>80.3</td>
<td>71.7</td>
</tr>
</tbody>
</table>
Plan

- **Weak Supervision NLP**
  - NLP, AI, software 2.0
  - Semantics as a foreign language
  - Unsupervised learning
  - Knowledge representation (symbolism)

- **Semantic Parsing Tasks**
  - WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines** (ACL 2017)
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)
  - RL vs MLE

- **Memory Augmented Policy Optimization** (NIPS 2018)
  - Experience replay (long term memory & optimal updating strategy)
  - Systematic exploration
  - Memory Weight Clipping (unbiased cold start strategy)

Access slides and join discussions at weakly-supervised-nlu google group
What is RL?

Directly Optimizing The Expected Reward

- **MLE** optimizes the log likelihood of target sequences
  \[
  J^{ML}(\theta) = \sum_{q} \log P(a_{0:T}^{\text{best}}(q)|q, \theta)
  \]

- **RL** optimizes the expected reward under a stochastic policy
  \[
  J^{RL}(\theta) = \sum_{q} \mathbb{E}_{P(a_{0:T}|q,\theta)}[R(q, a_{0:T})]
  \]

[Williams 1992]
[Sutton & Barto 1998]
Challenges of applying RL

- **Large search space (sparse rewards)**
  - Supervised pretraining (MLE)
  - Systematic exploration [Houthooft+ 2017]
  - Curiosity [Schmidhuber 1991][Pathak2017]

- **Credit assignment (delayed reward)**
  - Bootstrapping
    - E.g., AlphaGo uses a value function to estimate the future reward
  - Rollout n-steps

- **Train speed & stability (optimization)**
  - Trust region approaches (e.g., PPO)
  - Experience replay

```markdown
Our focus today
```
Efficiency challenge

- RL is still far from data efficient
  - E.g. the best learning algorithm (DeepMind RainbowDQN) “passes median human performance on 57 Atari games at about 18 million frames (around 90 hours) of gameplay, while most humans can pick up a game within a few minutes.”

- How to improve its efficiency?
Applying RL to NLP

● Benefits of RL
  ○ Weak supervision (e.g., expected answer, user clicks)
  ○ Directly optimizing the metric (e.g., F1, accuracy, BLEU etc.)
  ○ Work with structured hidden variables (e.g., logical forms/programs)

● Challenges with existing solutions
  ○ Large search space sparse reward often leads to slow and unstable training
  ○ Spurious reward often lead to biased solutions
**Training sample management issue**

- Too many low quality examples ⇒ slow training
- Boosting high reward experience ⇒ biased training
Complementary Learning Theory

Connections within and among neocortical areas (green) support gradual acquisition of structured knowledge through interleaved learning.

Bidirectional connections (blue) link neocortical representations to the hippocampus/MTL for storage, retrieval, and replay.

 Encoder

 Episodic memory

Rapid learning in connections within hippocampus (red) supports initial learning of arbitrary new information.

Record & Replay

Observations

Primary sensory and motor cortices

[McClelland+ 1995]
[Kumaran+ 2016]
Most of the past experience are not helpful for improving the current model
Mammals learn from “interesting” dreams

- In early 2000s, scientists discovered that animals have complex dreams and are able to retain and recall long sequences of events while they are asleep.

- Recent studies indicate that by consolidating memory traces with high emotional / motivational value, "sleep and dreaming may offer a neurobehavioral substrate for the offline ... learning”

Image courtesy of Kote on Drawception.com, 2012
Augment REINFORCE with Memory

Linear combination of maximum likelihood objective and expected return:

\[ \lambda \log \text{-likelihood on a top-k buffer} + (1 - \lambda) \text{expected return} \]

\[ \lambda \sum_{y \in \text{TopK}} \log p(y \mid x) + (1 - \lambda) \mathbb{E}_{\tilde{y} \sim p(y \mid x)} R(\tilde{y}) \]

- Not robust against spurious programs
- The composite objective is ad-hoc, and the gradient is biased
Spurious programs: right answer, wrong reason

Which nation won the most silver medal?

- **Correct program:**
  
  \[
  \text{(argmax rows “Silver”)} \\
  \text{(hop v1 “Nation”)}
  \]

- **Many spurious programs:**

  - \[
  \text{(argmax rows “Gold”)} \\
  \text{(hop v1 “Nation”)}
  \]

  - \[
  \text{(argmax rows “Bronze”)} \\
  \text{(hop v1 “Nation”)}
  \]

  - \[
  \text{(argmin rows “Rank”)} \\
  \text{(hop v1 “Nation”)}
  \]

  ...
Combating spurious rewards

- **Reinforce** a *rewarded experience* only if the model (current policy) also thinks that it is the right thing to do.
Memory Augmented Policy Optimization (MAPO)

Memory

Learner

Actor

Systematic Exploration

Good return?

Model checkpoint

NIPS [Liang+ 2018]
Lower the gradient variance without introducing bias

Given a memory buffer of (sequence, return) pairs: \( \mathcal{B} \equiv \left\{ \left( y^{(i)}, r^{(i)} \right) \right\}_{i=1}^{n} \), re-express expected return as,

\[
p(\mathcal{B}) \underbrace{\mathbb{E}_{p(\tilde{y})|\tilde{y} \in \mathcal{B}} R(\tilde{y})}_{\text{inside the buffer}} + (1 - p(\mathcal{B})) \underbrace{\mathbb{E}_{p(\tilde{y})|\tilde{y} \not\in \mathcal{B}} R(\tilde{y})}_{\text{outside the buffer}}
\]

- **Importance sampling**
  - Sample more frequently inside the buffer
  - Rejection sampling for samples outside the buffer.
Optimal Sample Allocation

Given that we want to apply stratified sampling to estimate the gradient of REINFORCE with baseline under 1/0 rewards. It can be shown that the optimal strategy is to allocate the **same number of samples** to **reward vs no reward** experiences.
Comparison of model update strategies

Correctness of the behavior

On-policy optimization (REINFORCE)

Iterative Maximum Likelihood (IML)

Maximum Marginal Likelihood (MML)

MAPO

Experiences & Reward

Model’s preference

NIPS [Liang+ 2018]
Memory Weight Clipping

- Policy gradient methods usually suffer from a **cold start problem**, because the model probabilities $P$ to good experiences are very small initially.

$$\nabla_{\theta} J^{RL}(\theta) = \sum_{q} \sum_{a_0:T} P(a_0:T|q, \theta) [R(q, a_0:T) - B(q)] \nabla_{\theta} \log P(a_0:T|q, \theta)$$

- We adopt a clipping mechanism to ensure that the buffer probability is larger than a threshold $\alpha$.

Sample $a_i^+ \sim \pi_{\theta}^{old}$ over $B_i$

$$w_i^+ \leftarrow \max(\pi_{\theta}^{old}(B_i), \alpha)$$

$$D \leftarrow D \cup (a_i^+, R(a_i^+), w_i^+)$$
Memory Weight Clipping

\[ \text{max}(P(B), \alpha) \]

\[ 1 - \text{max}(P(B), \alpha) \]

Trade off bias in the initial stage for faster training

NIPS [Liang+ 2018]
Training becomes less biased over time

NIPS [Liang+ 2018]
Comparison

- REINFORCE does not work at all
- MAPO is slower but less biased

The shaded area represents the standard deviation of the dev accuracy
### SOTA results with weak supervision

<table>
<thead>
<tr>
<th>Model</th>
<th>E.S.</th>
<th>Dev.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pasupat &amp; Liang (2015)[28]</td>
<td>-</td>
<td>37.0</td>
<td>37.1</td>
</tr>
<tr>
<td>Neelakantan et al. (2017)[26]</td>
<td>1</td>
<td>34.1</td>
<td>34.2</td>
</tr>
<tr>
<td>Neelakantan et al. (2017)[26]</td>
<td>15</td>
<td>37.5</td>
<td>37.7</td>
</tr>
<tr>
<td>Haug et al. (2017)[15]</td>
<td>1</td>
<td></td>
<td>34.8</td>
</tr>
<tr>
<td>Haug et al. (2017)[15]</td>
<td>15</td>
<td></td>
<td>38.7</td>
</tr>
<tr>
<td>Zhang et al. (2017)[51]</td>
<td>-</td>
<td>40.4</td>
<td>43.7</td>
</tr>
<tr>
<td>MAPO</td>
<td>1</td>
<td>42.7</td>
<td>43.8</td>
</tr>
<tr>
<td>MAPO (ensemble)</td>
<td>5</td>
<td></td>
<td>46.2</td>
</tr>
</tbody>
</table>

Table 3: Results on WikiTableQuestions. E.S. is the number of ensembles (if applicable).

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev.</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhong et al. (2017)[52]*</td>
<td>60.8</td>
<td>59.4</td>
</tr>
<tr>
<td>Wang et al. (2017)[40]*</td>
<td>67.1</td>
<td>66.8</td>
</tr>
<tr>
<td>Xu et al. (2017)[46]*</td>
<td>69.8</td>
<td>68.0</td>
</tr>
<tr>
<td>Huang et al. (2018)[18]*</td>
<td>68.3</td>
<td>68.0</td>
</tr>
<tr>
<td>Yu et al. (2018)[48]*</td>
<td>74.5</td>
<td>73.5</td>
</tr>
<tr>
<td>Sun et al. (2018)[38]*</td>
<td>75.1</td>
<td>74.6</td>
</tr>
<tr>
<td>Dong &amp; Lapata (2018)[12]*</td>
<td>79.0</td>
<td>78.5</td>
</tr>
<tr>
<td>MAPO</td>
<td>72.4</td>
<td>72.6</td>
</tr>
<tr>
<td>MAPO (ensemble of 5)</td>
<td></td>
<td>74.9</td>
</tr>
</tbody>
</table>

Table 4: Results on WikiSQL. *All other methods use question-program pairs as strong supervision, while MAPO only uses question-answer pairs as weak supervision.
Scale up: Distributed Actor-Learner architecture

[Scale up diagram with nodes for Train set shard 1, Envs 1, Actor 1, Sample queue, Checkpoint queue, Learner, etc.]

[References to Liang et al, 2017; Espeholt et al, 2018; Liang et al, 2018]
Thanks!

- **Weak Supervision NLP**
  - NLP, AI, software 2.0
  - Semantics as a foreign language
  - Unsupervised learning
  - Knowledge representation (symbolism)

- **Semantic Parsing Tasks**
  - WebQuestionsSP, WikiTableQuestions

- **Neural Symbolic Machines** (ACL 2017)
  - Compositionality (short term memory)
  - Scalable KB inference (symbolism)
  - RL vs MLE

- **Memory Augmented Policy Optimization** (NIPS 2018)
  - Experience replay (long term memory & optimal updating strategy)
  - Systematic exploration
  - Memory Weight Clipping (unbiased cold start strategy)

Access slides and join discussions at weakly-supervised-nlu google group