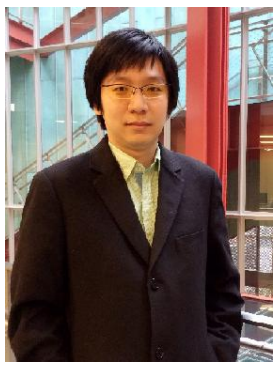
Learning to Search Better than Your Teacher

Akshay Krishnamurthy



Kai-Wei Chang

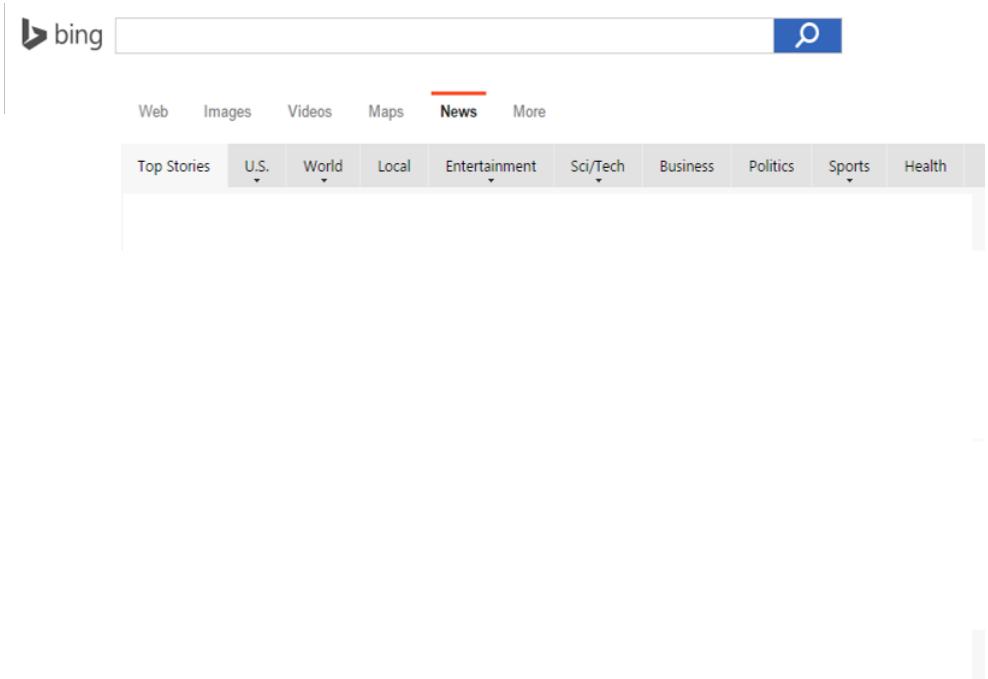
Alekh Agarwal

Hal Daumé III

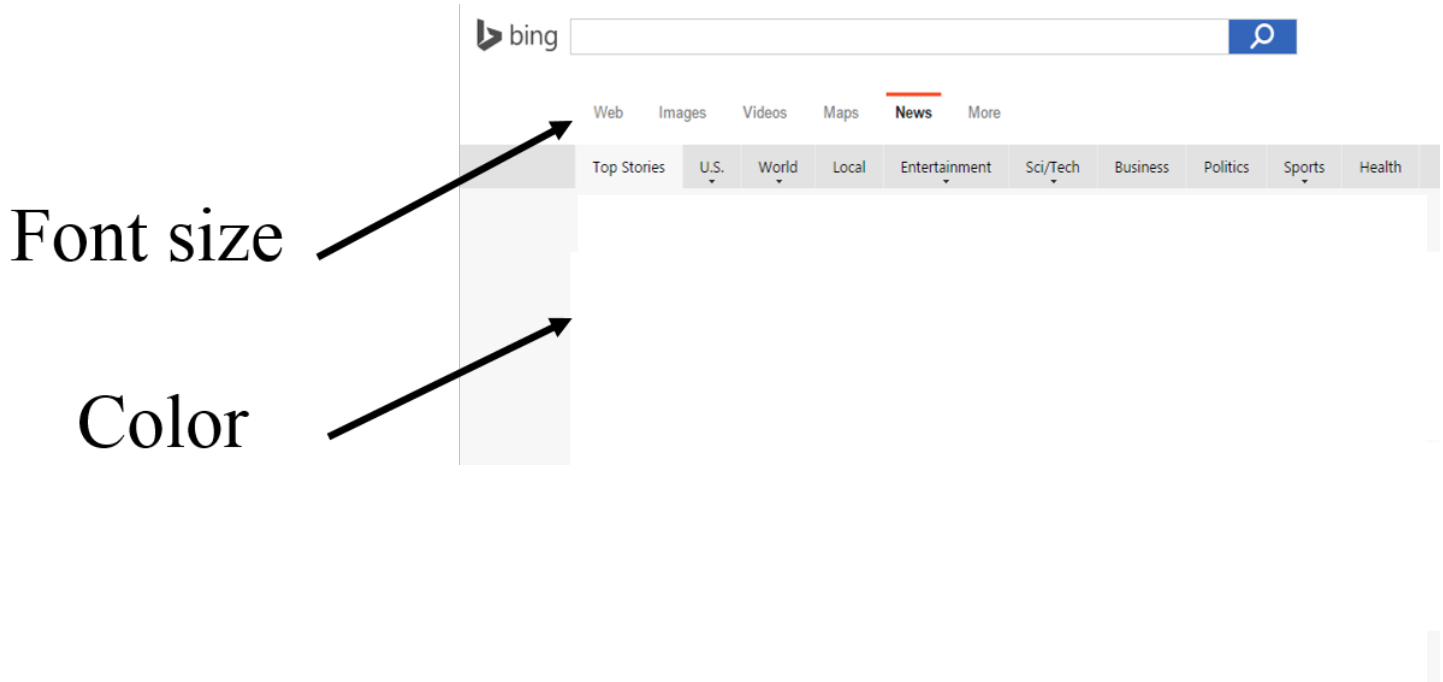
John Langford

Microsoft Research, NY.

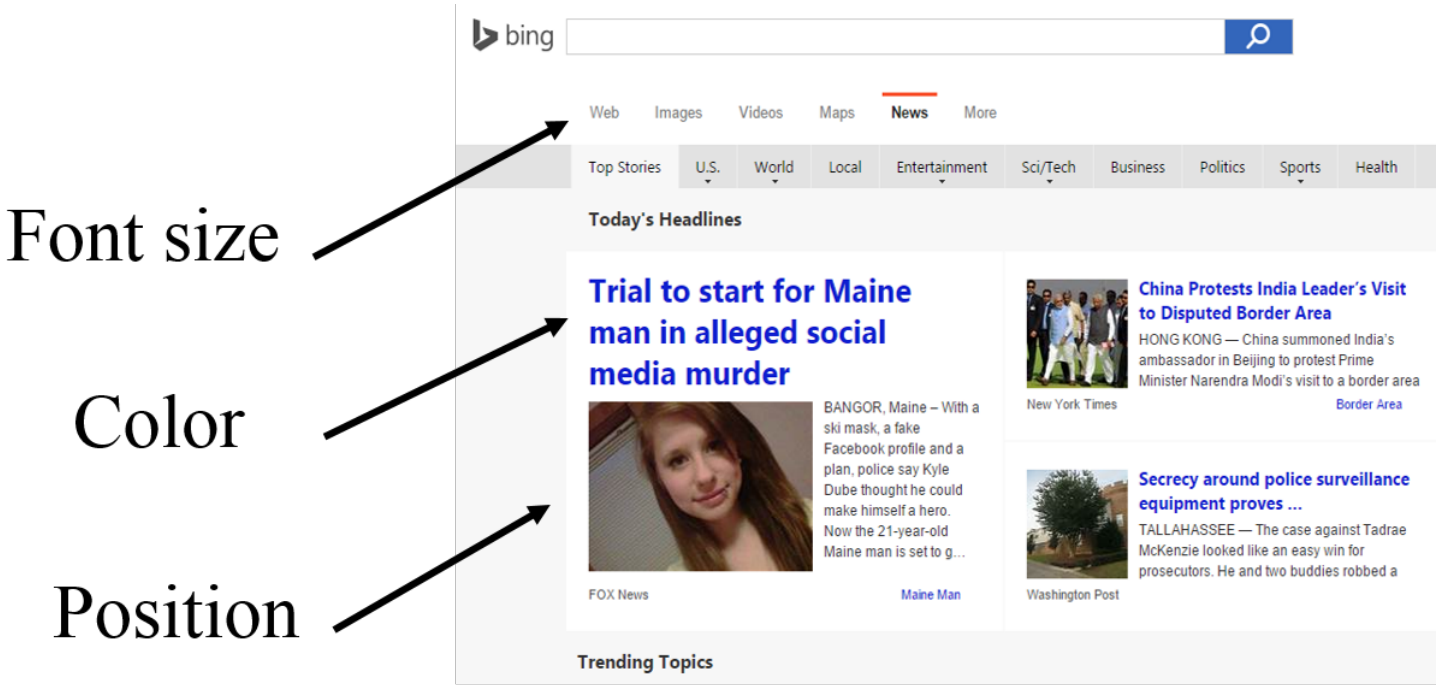
Structured Contextual Bandit



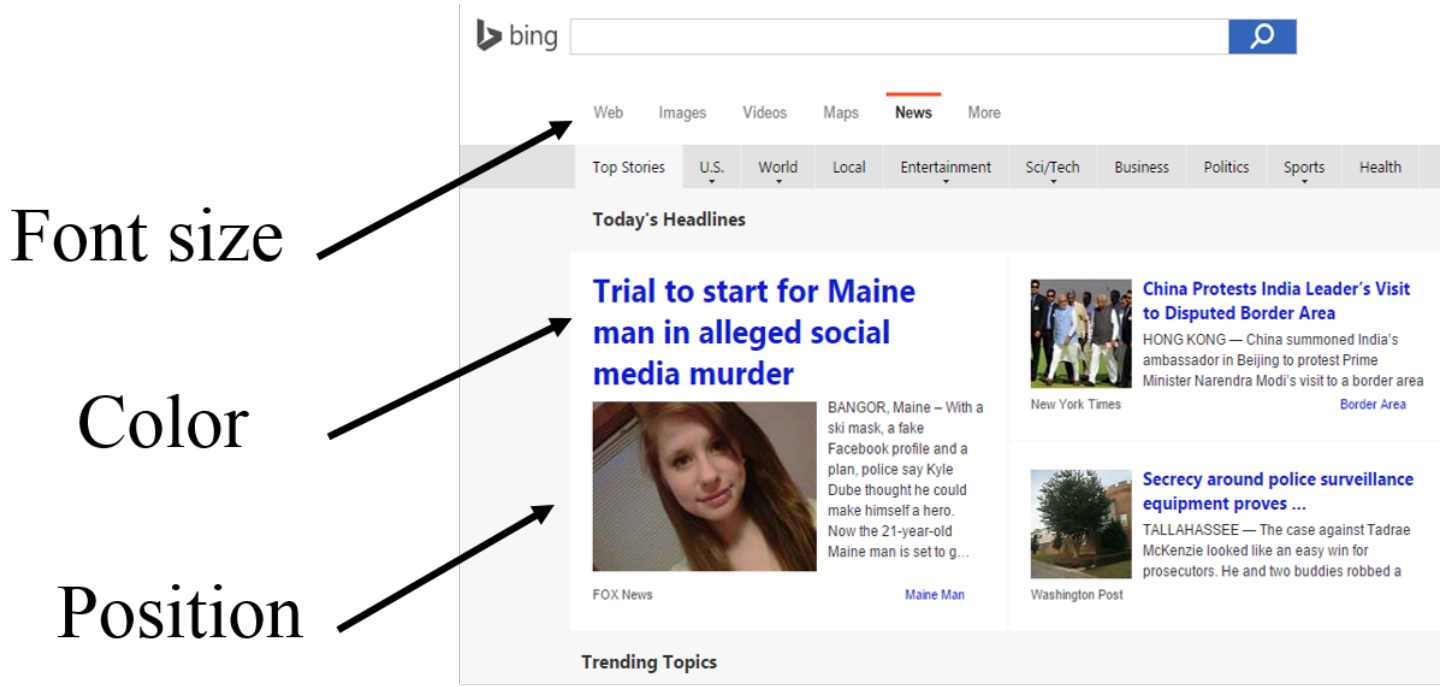
Structured Contextual Bandit



Structured Contextual Bandit

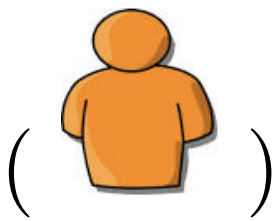


Structured Contextual Bandit



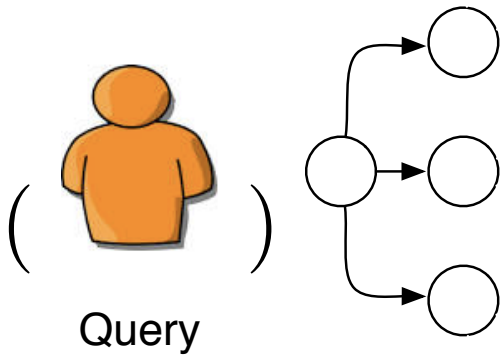
- Loss of a single structured label can be observed.

A Search Problem

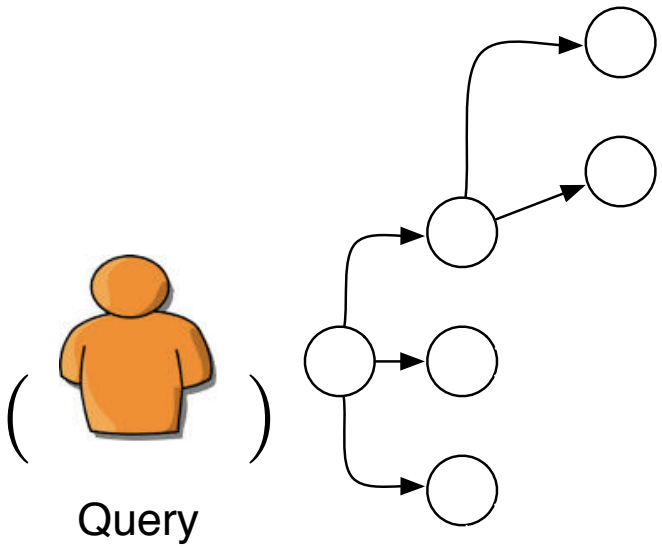


Query

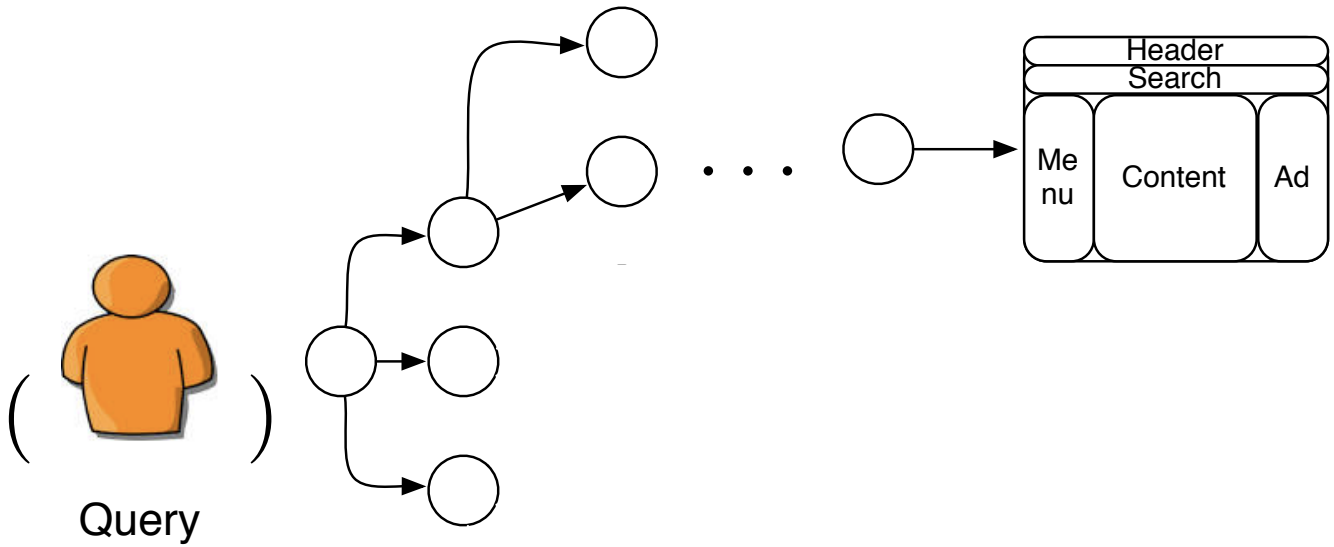
A Search Problem



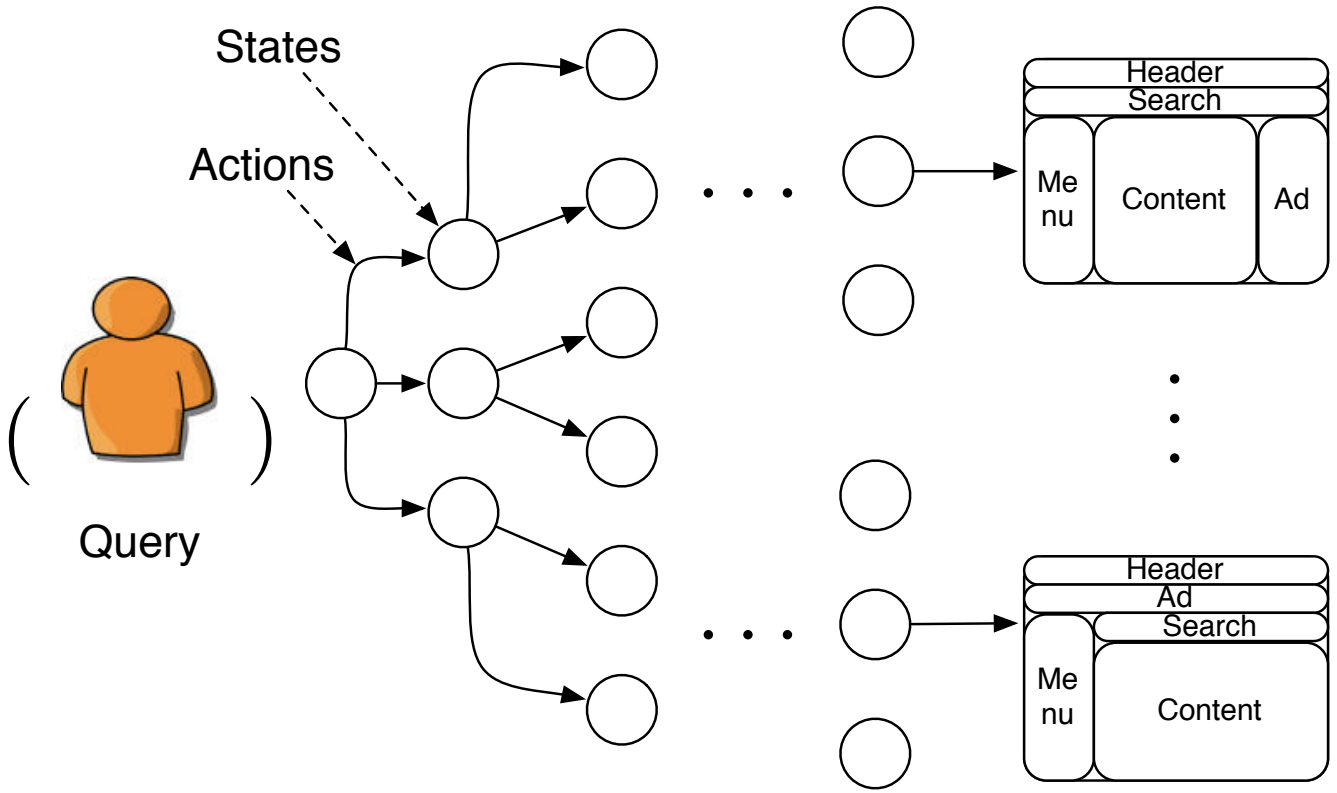
A Search Problem



A Search Problem

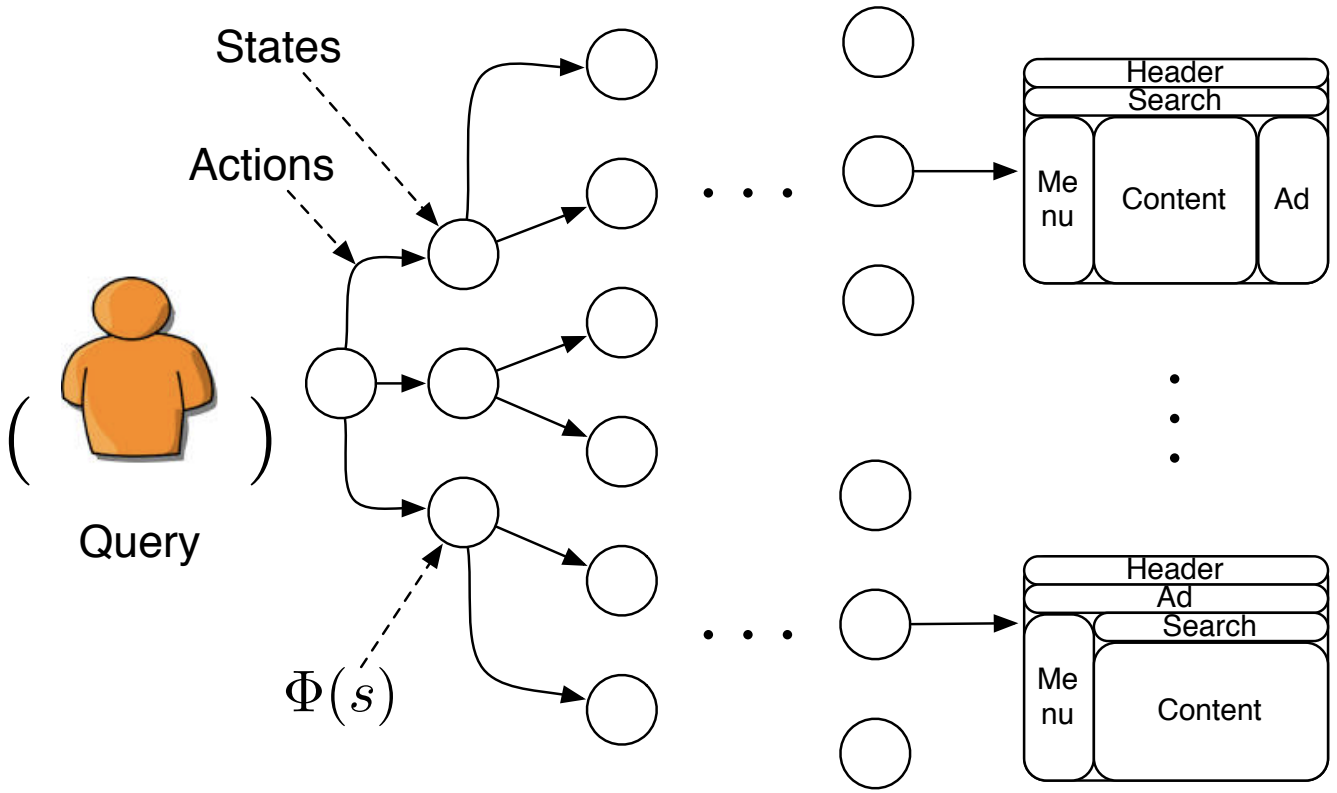


A Search Problem



Convert SCB into search problem (search space + actions).

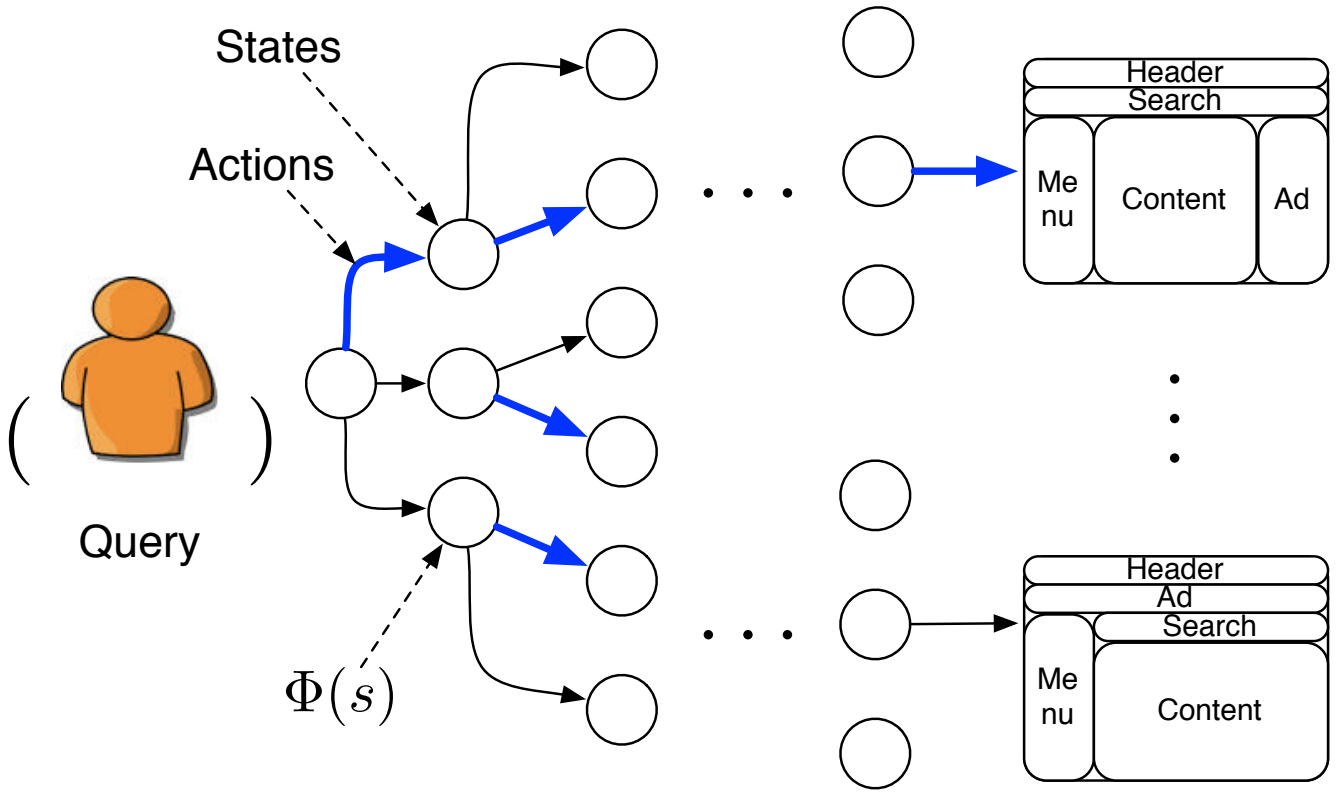
A Search Problem



Define structured features over each state.

- Learn policy mapping features to actions.

A Search Problem



Use status quo system as reference policy (Ref).

- Goal is to improve on Ref.

Connection with Learning to Search

	L2S	SCB

Connection with Learning to Search

	L2S	SCB
Search Space	States, Actions, Features	

Connection with Learning to Search

	L2S	SCB
Search Space	States, Actions, Features	
Feedback	Labels	Single Loss

Connection with Learning to Search

	L2S	SCB
Search Space	States, Actions, Features	
Feedback	Labels	Single Loss
Ref Quality	Optimal	Can be bad

Connection with Learning to Search

	L2S	SCB
Search Space	States, Actions, Features	
Feedback	Labels	Single Loss
Ref Quality	Optimal	Can be bad
Ref Availability	Training only	Train and Test

Connection with Learning to Search

	L2S	SCB
Search Space	States, Actions, Features	
Feedback	Labels	Single Loss
Ref Quality	Optimal	Can be bad
Ref Availability	Training only	Train and Test
Goal	Imitate Ref	Improve upon Ref

Connection with Learning to Search

	L2S	SCB
Search Space	States, Actions, Features	
Feedback	Labels	Single Loss
Ref Quality	Optimal	Can be bad
Ref Availability	Training only	Train and Test
Goal	Imitate Ref	Improve upon Ref

Existing L2S algorithms give:

$$L(\pi) \leq L(\pi^{\text{ref}}) + o(1)$$

Main Goal

Learning to Search with:

1. A suboptimal reference $\Rightarrow$ Learn policy that improves on Ref.

Main Goal

Learning to Search with:

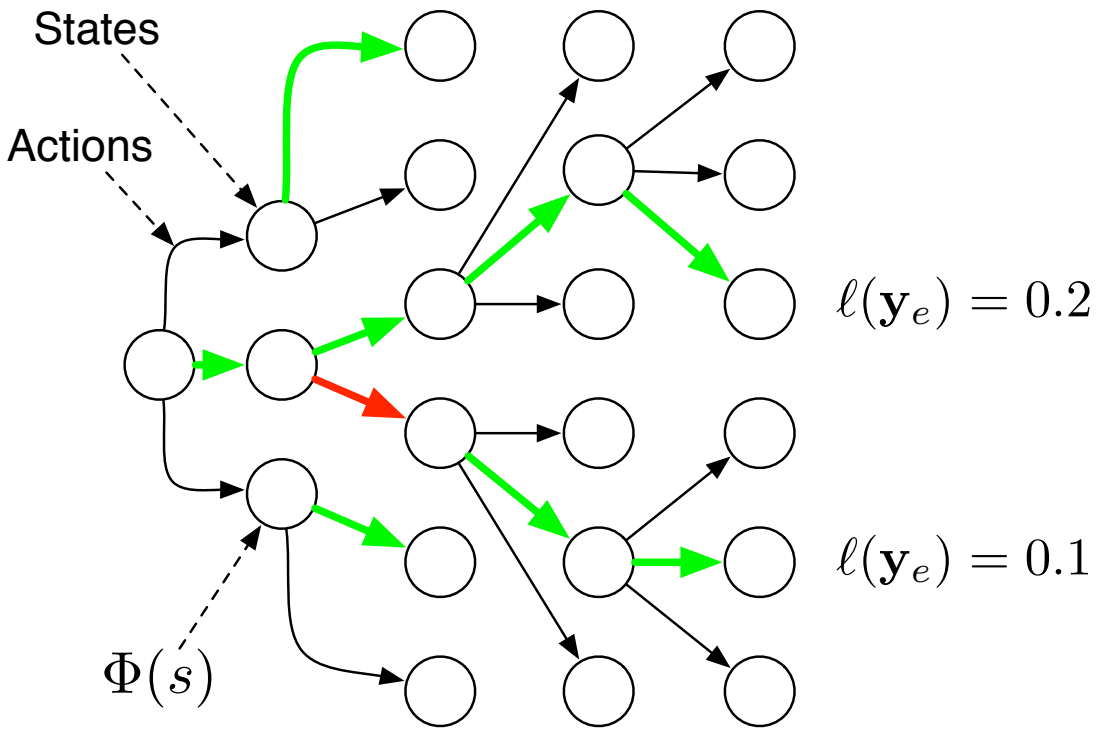
1. A suboptimal reference $\Rightarrow$ Learn policy that improves on Ref.
2. Partial feedback.

Disederata

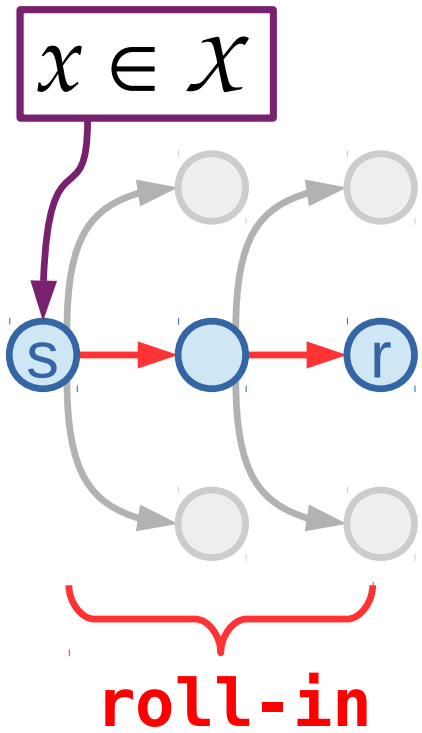
- Compete with Ref.
  - Global optimality if Ref is optimal and realizable.

Disederata

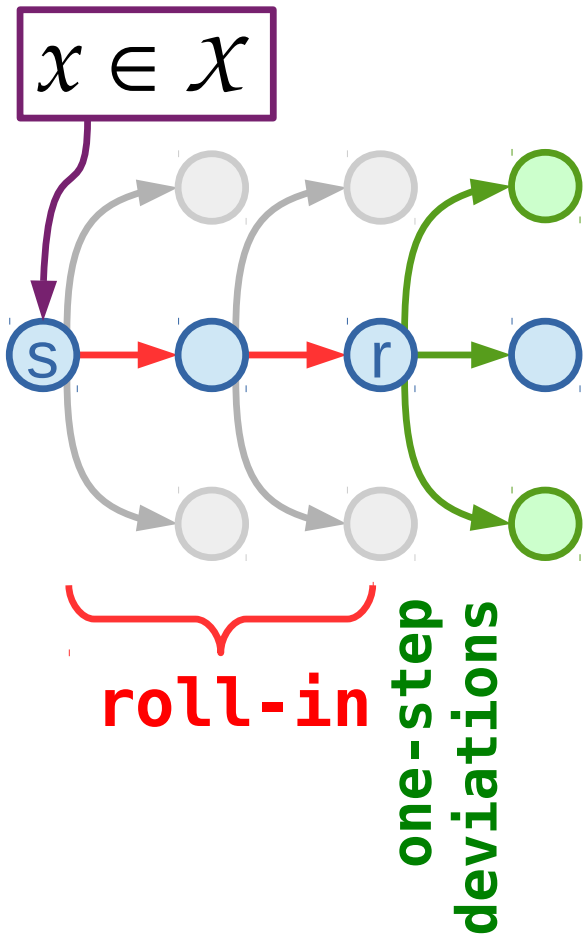
- Compete with Ref.
  - Global optimality if Ref is optimal and realizable.
- Local optimality.
  - Compete with your own one-step deviations.



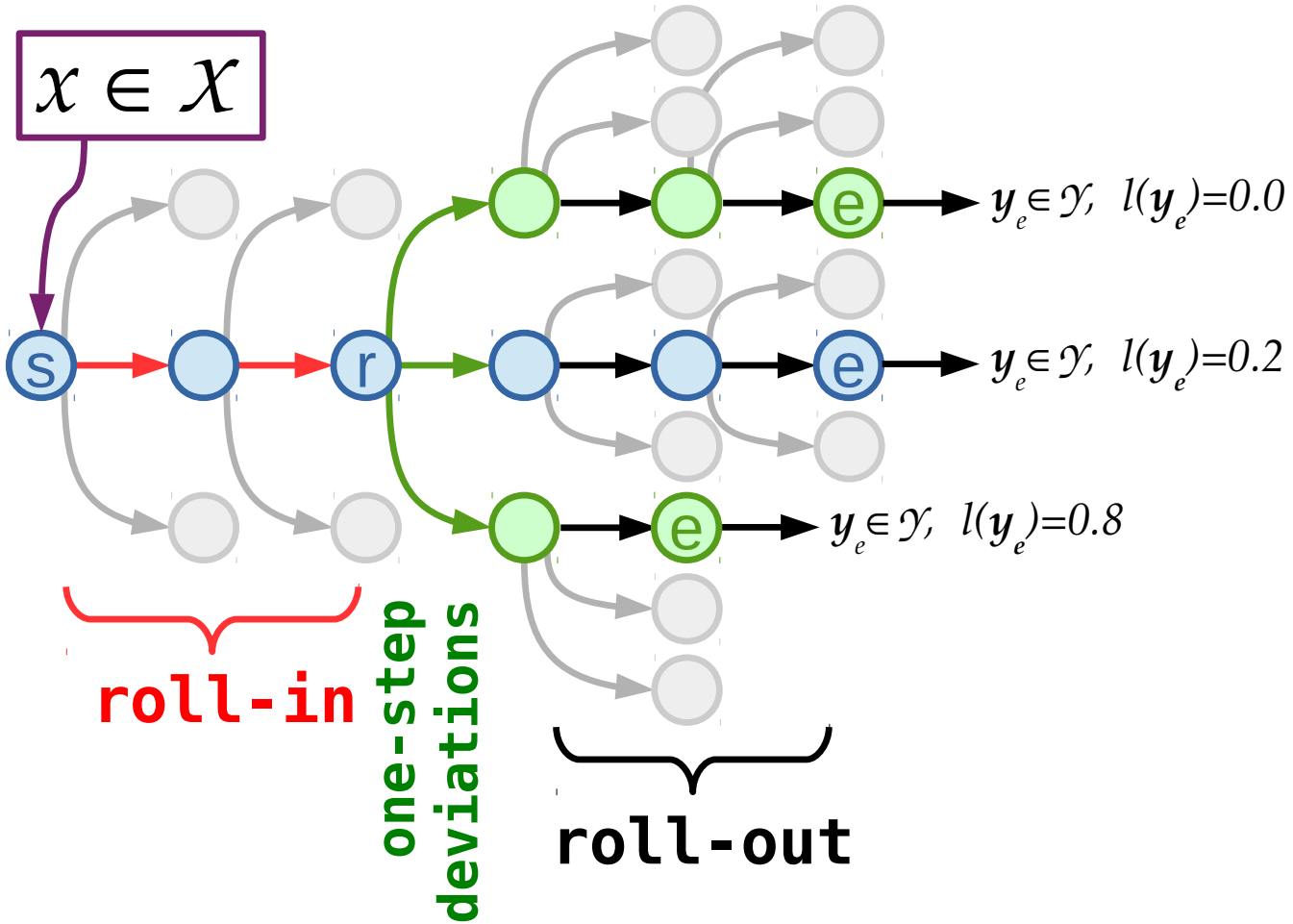
Exploration in Search Space



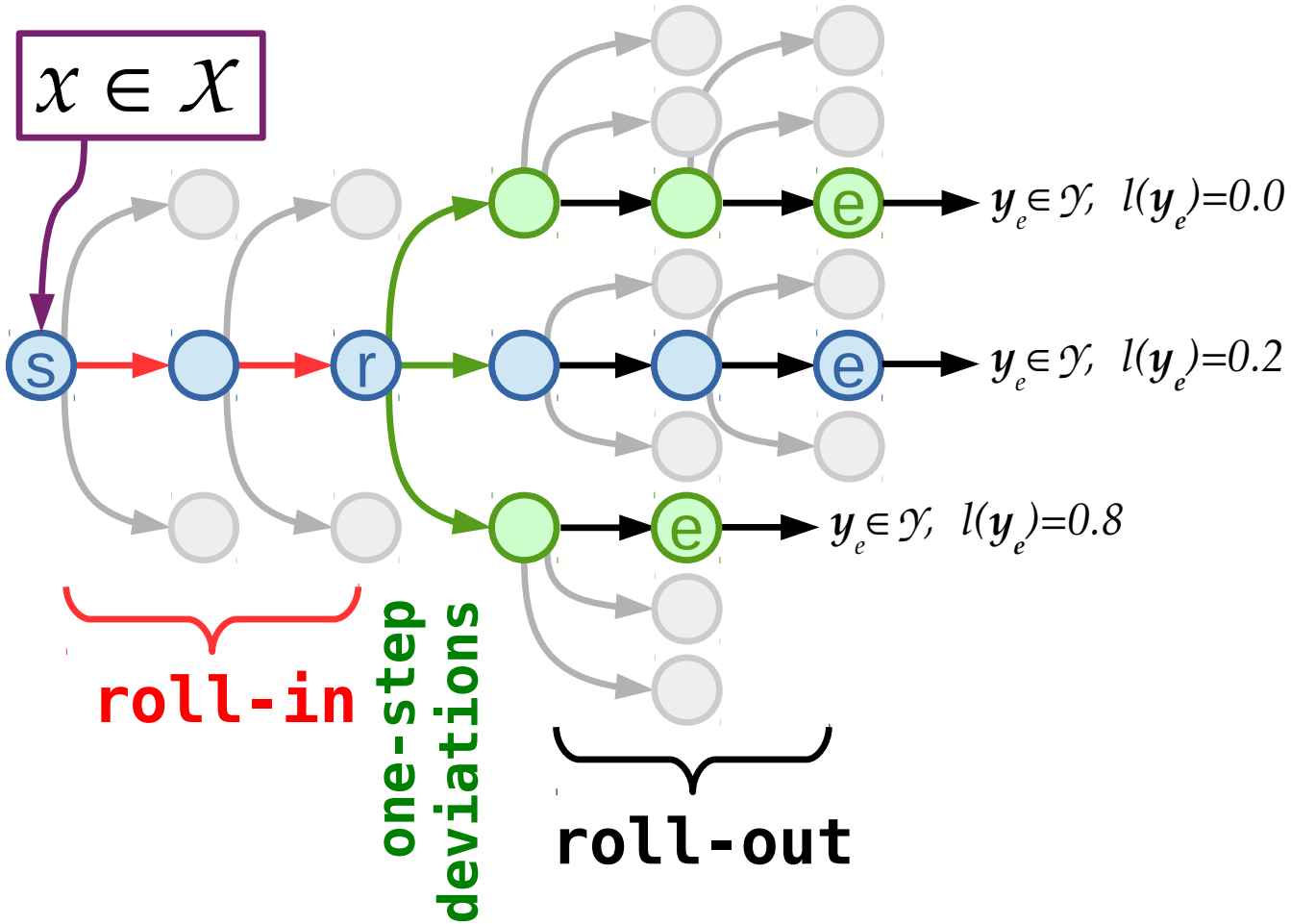
Exploration in Search Space



Exploration in Search Space



Exploration in Search Space



- Roll-in choice affects what states we train on.
- Roll-out choice affects how we score actions.

Effect of Roll-in and Roll-out Policies

roll-out → ↓ roll-in	Reference	Half/Half	Learned
Reference	Inconsistent		
Learned	No local opt	Good	RL

Effect of Roll-in and Roll-out Policies

roll-out → / ↓ roll-in	Reference	Half/Half	Learned
Reference	**Inconsistent**		
Learned	No local opt	Good	RL

Theorem

Roll-in with Ref can generate a model with unbounded structured regret but zero cost-sensitive regret.

- States trained on are not representative of those seen at prediction time.

Effect of Roll-in and Roll-out Policies

roll-out → / ↓ roll-in	Reference	Half/Half	Learned
Reference	Inconsistent		
Learned	**No local opt**	Good	RL

Theorem

Roll-out with Ref may cause the learned policy to fail to converge to a local optimum if the reference policy is suboptimal.

- Causes poor assessment of comparison policies.

Effect of Roll-in and Roll-out Policies

roll-out → / ↓ roll-in	Reference	Half/Half	Learned
Reference	Inconsistent		
Learned	No local opt	Good	**RL**

Theorem

Roll-in and roll-out with the learned policy ignores Ref and is equivalent to reinforcement learning.

Effect of Roll-in and Roll-out Policies

roll-out → / ↓ roll-in	Reference	Half/Half	Learned
Reference	Inconsistent		
Learned	No local opt	**Good**	RL

Theorem

LOLS minimizes a combination of regret to Ref and regret to its own one-step deviations.

LOLS Regret Bound

Theorem

LOLS minimizes a combination of regret to Ref and regret to its own one-step deviations.

$$\frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{ref}})\right)}_{\text{Regret to Ref}} + \frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{dev}})\right)}_{\text{Regret to devs}} \text{ is small.}$$

LOLS Regret Bound

Theorem

LOLS minimizes a combination of regret to Ref and regret to its own one-step deviations.

$$\frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{ref}})\right)}_{\text{Regret to Ref}} + \frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{dev}})\right)}_{\geq 0} \text{ is small.}$$

- Competes with Ref.

LOLS Regret Bound

Theorem

LOLS minimizes a combination of regret to Ref and regret to its own one-step deviations.

$$\frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{ref}})\right)}_{\geq 0} + \frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{dev}})\right)}_{\text{Regret to devs}} \text{ is small.}$$

- Competes with Ref.
- Locally optimal when Ref is optimal (even if unrealizable).

LOLS Regret Bound

Theorem

LOLS minimizes a combination of regret to Ref and regret to its own one-step deviations.

$$\frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{ref}})\right)}_{\text{Regret to Ref}} + \frac{1}{2}\underbrace{\left(L(\hat{\pi}) - L(\pi^{\text{dev}})\right)}_{\text{Regret to devs}} \text{ is small.}$$

- Competes with Ref.
- Locally optimal when Ref is optimal (even if unrealizable).
- If Ref suboptimal, either locally optimal or better than Ref.

Local Optimality is Hard

Finding local optimum could be hard without further assumptions.

Theorem

Can require $\Omega(2^T)$ cost-sensitive classification examples to reach local optimum.

T is the number of decisions per example.

Dependency Parsing

Find the dependency structure of words in a sentence.

roll-out → / ↓ roll-in	**Reference**	**Mixture**	**Learned**
Reference is optimal			
Reference	87.2	89.7	88.2
Learned	**90.7**	90.5	86.9

Dependency Parsing

Find the dependency structure of words in a sentence.

roll-out → / ↓ roll-in	**Reference**	**Mixture**	**Learned**
Reference is optimal			
Reference	87.2	89.7	88.2
Learned	**90.7**	90.5	86.9
Reference is suboptimal			
Reference	83.3	87.2	81.6
Learned	87.1	**90.2**	86.8

Dependency Parsing

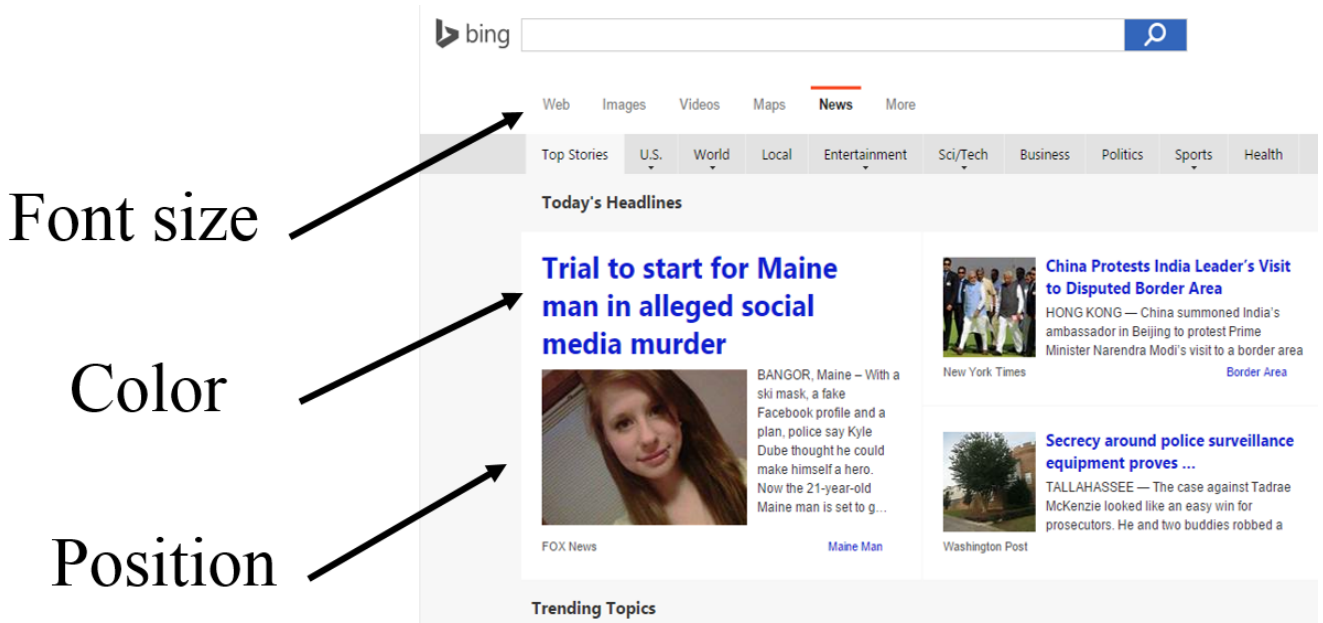
Find the dependency structure of words in a sentence.

roll-out → / ↓ roll-in	**Reference**	**Mixture**	**Learned**
Reference is optimal			
Reference	87.2	89.7	88.2
Learned	**90.7**	90.5	86.9
Reference is suboptimal			
Reference	83.3	87.2	81.6
Learned	87.1	**90.2**	86.8
Reference is bad			
Reference	68.7	65.4	66.7
Learned	75.8	**89.4**	87.5

LOLS always good, even with suboptimal Ref.

Structured Contextual Bandit

- Loss of a single structured label can be observed.
- Reference policy is not optimal under this setting.



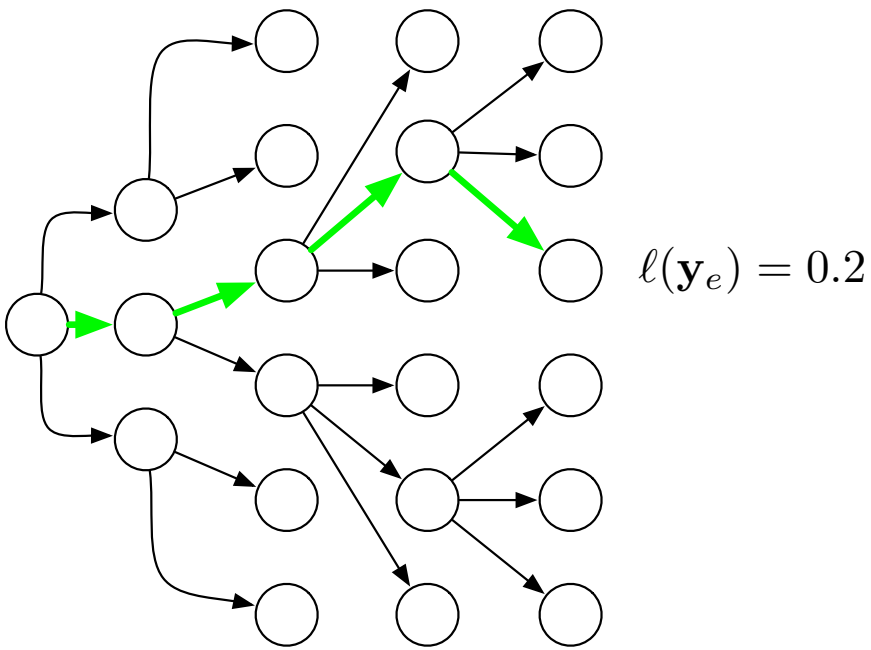
Structured Contextual Bandit

We adapt an ϵ-greedy algorithm
Upon receiving an example:

Structured Contextual Bandit

We adapt an ϵ-greedy algorithm
Upon receiving an example:

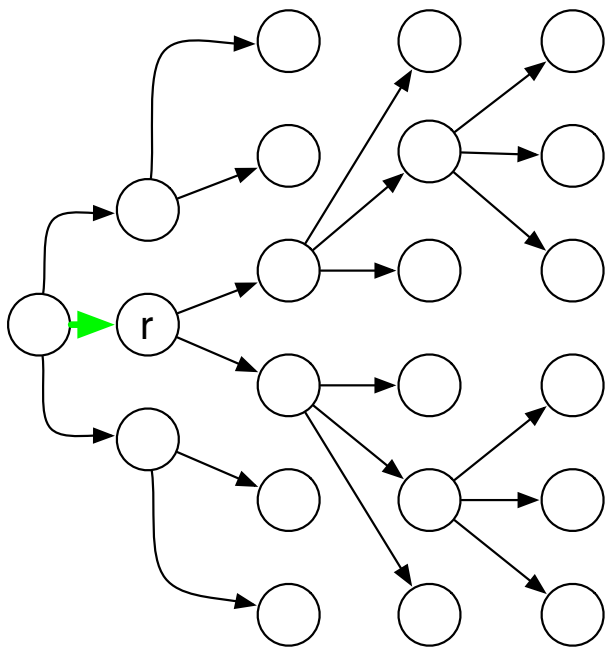
- w/ prob. $1 - \epsilon$: follow the current policy



Structured Contextual Bandit

We adapt an ϵ-greedy algorithm
Upon receiving an example:

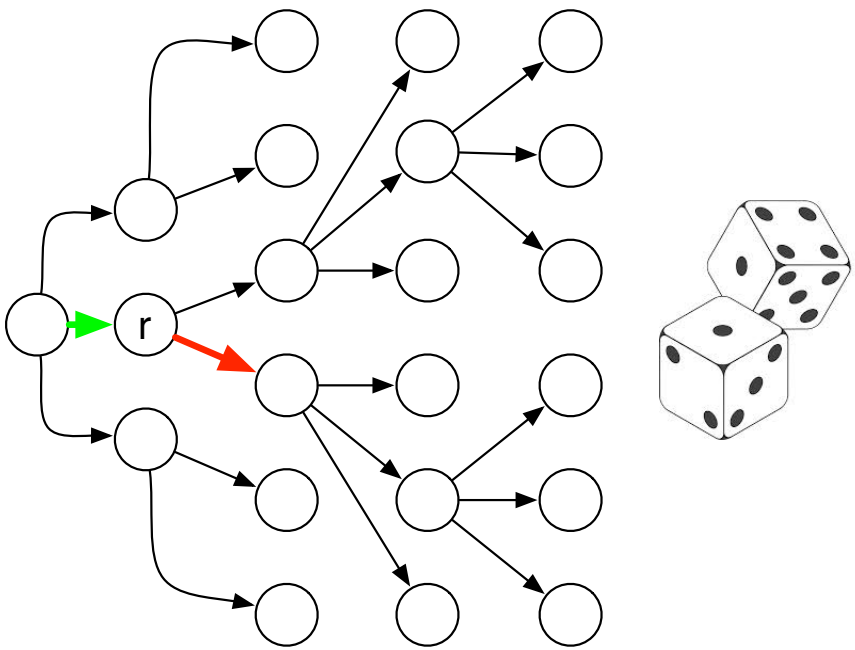
- w/ prob. $1 - \epsilon$: follow the current policy
- w/ prob. ϵ: perform a randomized LOLS update.



Structured Contextual Bandit

We adapt an ϵ-greedy algorithm
Upon receiving an example:

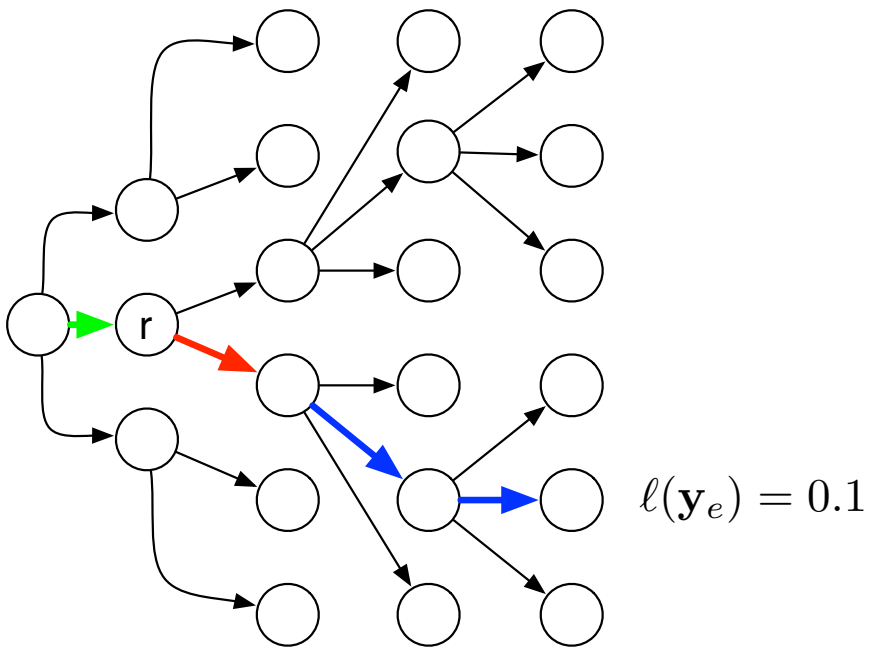
- w/ prob. $1 - \epsilon$: follow the current policy
- w/ prob. ϵ: perform a randomized LOLS update.



Structured Contextual Bandit

We adapt an ϵ-greedy algorithm
Upon receiving an example:

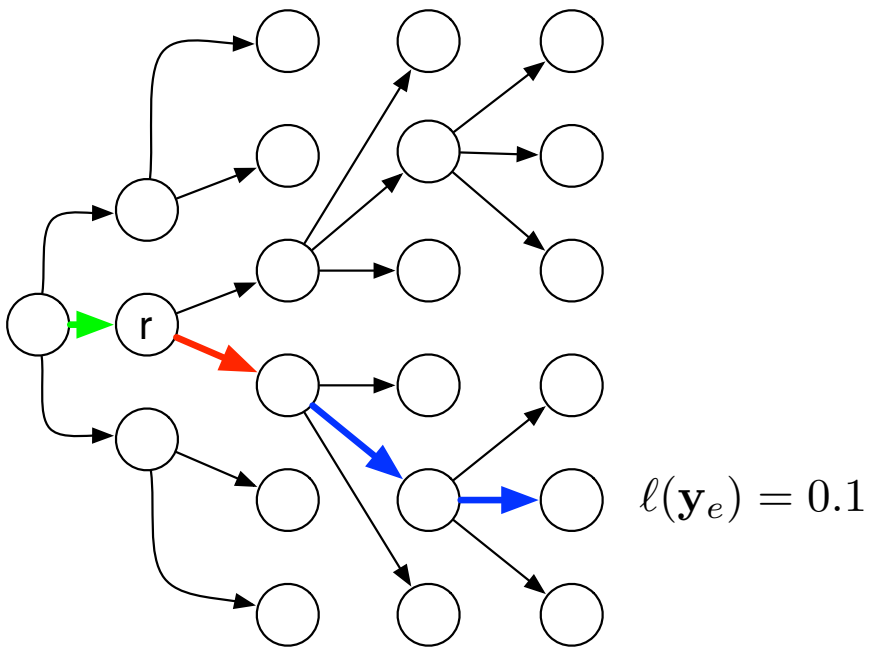
- w/ prob. $1 - \epsilon$: follow the current policy
- w/ prob. ϵ: perform a randomized LOLS update.



Structured Contextual Bandit

We adapt an ϵ-greedy algorithm
Upon receiving an example:

- w/ prob. $1 - \epsilon$: follow the current policy
- w/ prob. ϵ: perform a randomized LOLS update.



Regret against ref and deviations is still bounded.

Our Contributions

1. What is a good definition of local optimality?
   - Competitive with one-step deviations.

Our Contributions

1. What is a good definition of local optimality?
   - Competitive with one-step deviations.
2. Can L2S work even when the reference is poor?
   - Yes. LOLS competes with ref and one-step deviations.

Our Contributions

1. What is a good definition of local optimality?
   - Competitive with one-step deviations.
2. Can L2S work even when the reference is poor?
   - Yes. LOLS competes with ref and one-step deviations.
3. Can L2S work in a partial information setting?
   - Yes. ϵ-Greedy style algorithm.

Our Contributions

1. What is a good definition of local optimality?
   - Competitive with one-step deviations.
2. Can L2S work even when the reference is poor?
   - Yes. LOLS competes with ref and one-step deviations.
3. Can L2S work in a partial information setting?
   - Yes. ϵ-Greedy style algorithm.

Thanks!