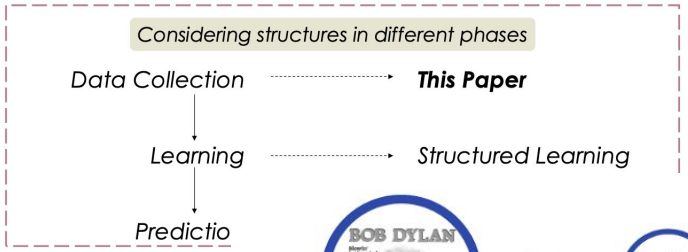
Partial Or Complete, That Is The Question

Authors: Qiang Ning, Hangfeng He, Chuchu Fan, Dan Roth
Venue: NAACL (2019)

Presenter: Xingyu Fu

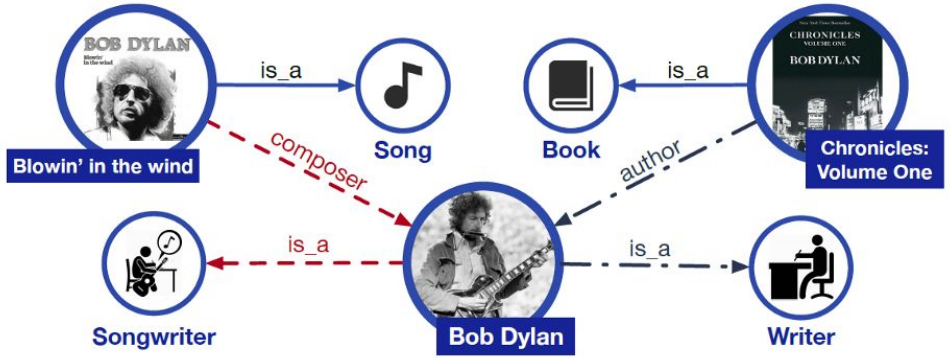
Background: Structured Data



[Image from author's poster.]

Structured Data: a prediction typically involves assigning values to multiple variable that are interrelated.

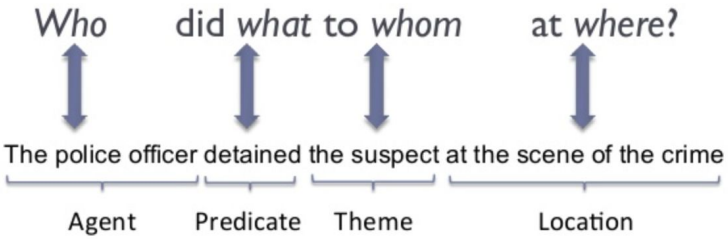
nantic Role Labeling, …



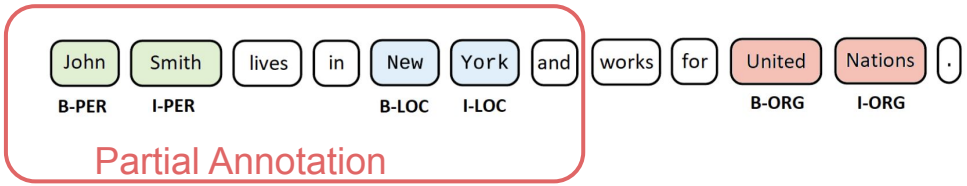
Bob Dylan wrote ***Blowin' in the Wind*** in 1962, and wrote ***Chronicles: Volume One*** in 2004.

Background: Structured Data Annotation

Semantic Role Labeling



Sequence Labeling
(NER here)



Existing annotation: Complete structures, complex and expensive.

However, we may not have enough budget …. And sometimes, we may not have complete structures….

Is Partial Annotation a compromise?

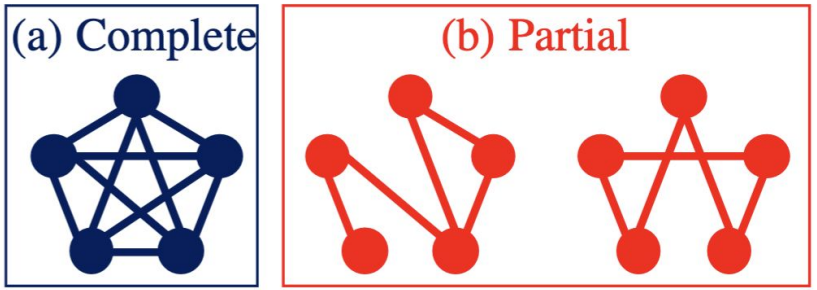
Common perception: Partial structures are low quality, could hurt the learning process.

Claim: Structures consist of interdependent sets of variables. Partly annotating each structure may provide the same level of supervision, within fixed budget.

★ **Important Assumptions**: **uniform cost** over individual annotations (E.g. Each edge's cost in graph labeling.)

Motivation: Individual instances put restrictions on others.

E.g. Edge Annotation (k edges):



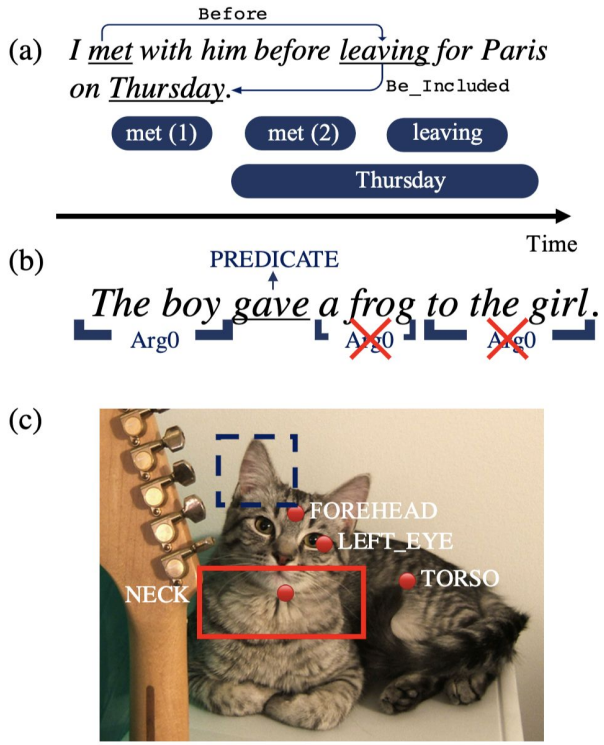
Example of Partial Annotation

Motivation: Individual instances put restrictions on others.

(a) The temporal relation between *met* and Thursday has to be BEFORE (“met (1)”) or BE_INCLUDED (“met (2)”).

(b) The argument roles of a frog and to the girl cannot be ARG_0 anymore.

(c) Given the position of the cat’s FOREHEAD and LEFT EYE, a rough estimate of its NECK can be the red solid box rather than the blue dashed box.



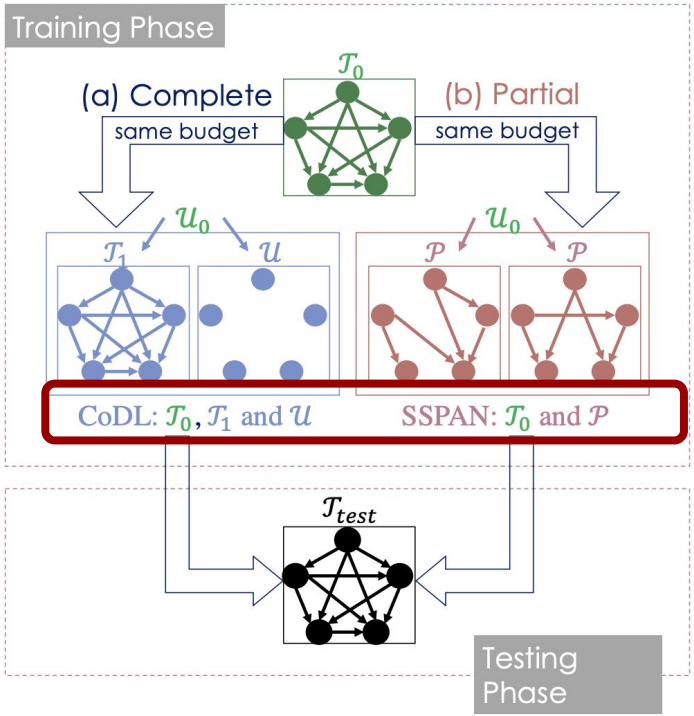
Compare: Partial or Complete?

★ **Important Assumptions**: equal cost for Complete /Partial Annotation.

1. Use same budget to get complete annotations and partial annotations.
2. Learn a model for each annotation, compare the performance on same unseen and complete test set.

- Complete Annotation: structures full / empty.
- Partial Annotation: Only partial structures.

T , P, and U denote complete, partial, and empty structures, respectively. T0 is a small but complete dataset for good initialization.



[Image comes from the Author's poster.]

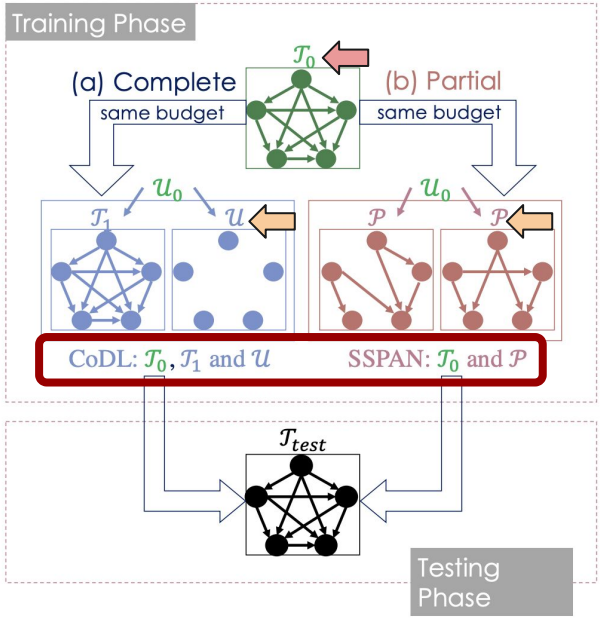
Compare: 1. Build Partial Annotation

Method: Early stopping partial annotation (ESPA)

Operation:

Randomly picks up instances to label in the beginning, and stops before a structure is completed.

Compare: 2. Training Algorithms for Complete/Partial



Input: $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, $\mathcal{P} = \{(\mathbf{x}_i, \mathbf{a}_i)\}_{i=N+1}^{N+M}$

Initialize $\mathcal{H} = \text{LEARN}(\mathcal{T})$

while *convergen*

  $\tilde{\mathcal{P}} = \emptyset$

  foreach $(\mathbf{x}_i, \mathbf{a}_i) \in \mathcal{P}$

    $\hat{\mathbf{y}}_i = \text{INFERENCE}(\mathbf{x}_i; \mathcal{H})$, such that

      ◇ $\hat{\mathbf{y}}_i \in C(\mathcal{Y}^d)$

      ◇ $\hat{y}_{i,j} = a_{i,j}, \forall a_{i,j} \neq \sqcap$ ★

    $\tilde{\mathcal{P}} = \tilde{\mathcal{P}} \cup \{(\mathbf{x}_i, \hat{\mathbf{y}}_i)\}$

  $\mathcal{H} = \text{LEARN}(\mathcal{T} + \tilde{\mathcal{P}})$

return $\mathcal{H}$

Good initialization.

Complete version of P

Self Learning

complete missing annotations

Compare: 2. Training Algorithms for Complete/Partial

Complete: CoDL, constraint-driven learning; can be seen as "structured self-learning" [ACL'07] .

Partial: SSPAN, extension of CoDL [*SEM'18].

Input: $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}, \mathcal{P} = \{(\mathbf{x}_i, \mathbf{a}_i)\}_{i=N+1}^{N+M}$

1 Initialize $\mathcal{H} = \textsc{Learn}(\mathcal{T})$

2 **while** *convergence criteria not satisfied* **do**

3 $\quad \tilde{\mathcal{P}} = \emptyset$

4 $\quad$ **foreach** $(\mathbf{x}_i, \mathbf{a}_i) \in \mathcal{P}$ **do**

5 $\quad\quad \hat{\mathbf{y}}_i = \textsc{Inference}(\mathbf{x}_i; \mathcal{H})$, such that

6 $\quad\quad\quad \diamond\ \hat{\mathbf{y}}_i \in C(\mathcal{Y}^d)$

7 $\quad\quad\quad \diamond\ \hat{y}_{i,j} = a_{i,j}, \forall a_{i,j} \neq \sqcap$ ⭐

8 $\quad\quad \tilde{\mathcal{P}} = \tilde{\mathcal{P}} \cup \{(\mathbf{x}_i, \hat{\mathbf{y}}_i)\}$

9 $\quad \mathcal{H} = \textsc{Learn}(\mathcal{T} + \tilde{\mathcal{P}})$

10 **return** $\mathcal{H}$

Self-Learning

Line 6: inference follows constraints.
Line 7 enforces partial annotations.

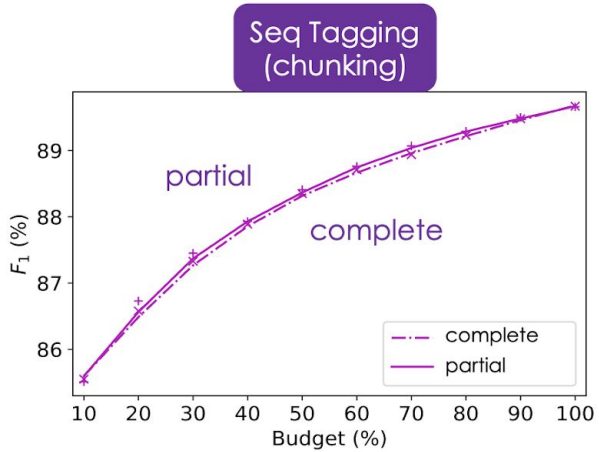
LEARN & INFERENCE use existing models.

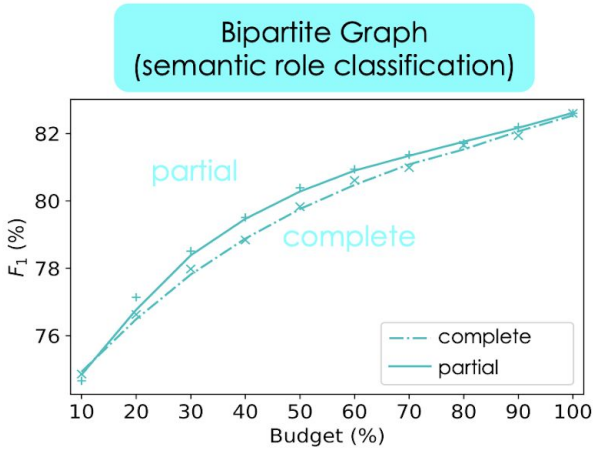
Note: Without ⭐, SSPAN goes back to CoDL.

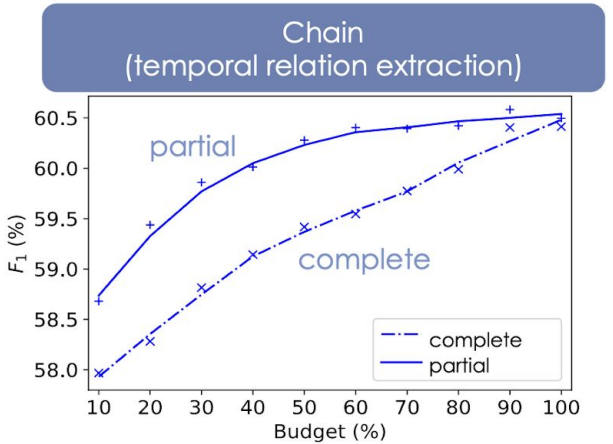
Experiment: Tasks

- **Chunking:** Similar to NER. It identifies text chunks in a sentence, such as noun phrases (NP), verb phrases (VP), etc. That is, Labels = {B-NP, I-NP, B-VP, I-VP, . . . , O}. (B(egin), I(nside), and O(utside)).
  - **Structural Constraint**: O(utside) cannot be immediately followed by I(nside).
  - Dataset: CoNLL-2000 shared task (Tjong Kim Sang and Buchholz, 2000).
  - Model: Chunker provided in CogCompNLP(Khashabi et al., 2018), sparse averaged perceptron as LEARN, and the INFERENCE is described in (Punyakanok and Roth, 2001).
- **Semantic Role Classification (SRC)** is a subtask of SRL (Who did What to Whom at Where). It assumes gold predicates and argument chunks, and only classifies the semantic role of each argument. So it is an assignment problem.
  - **Structure Constraint**: Each argument has exactly one semantic role, and the same role cannot appear twice for a single verb.
  - Dataset: Wall Street Journal (WSJ) part of Penn TreeBank III (Marcus et al., 1993).
  - Model: Adopt SRL system in CogCompNLP , sparse averaged perceptron as LEARN, ILP as INFERENCE.
- **Temporal relations (TempRel)** are a type of important relations representing the temporal ordering of events. That is to answer questions like which event happens earlier or later in time.
  - **Structure Constraint**: Transitivity Constraints. if A is before B and B is also before C, then A must be before C.
  - Dataset: MATRES dataset (Ning et al., 2018b)
  - Model: Features chosen following CogCompTime(Ning et al., 2018d), sparse averaged perceptron as LEARN, ILP as INFERENCE.

Experiment: Results

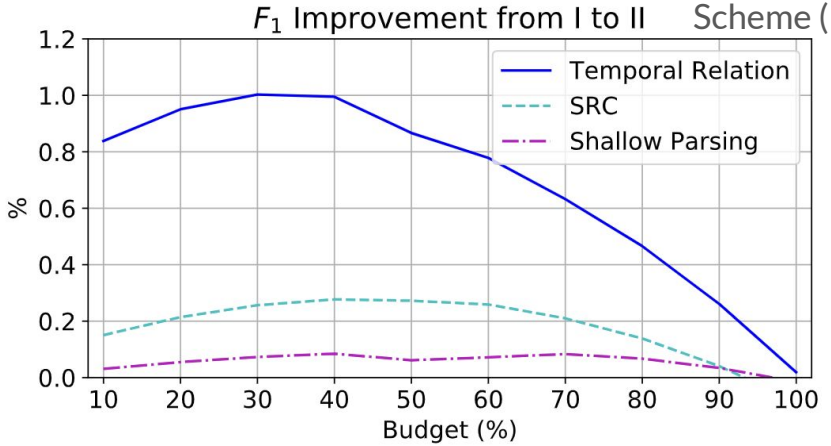






- The improvement of F1 brought by Partial annotation: chunking < SRC < Temporal Relation.

Experiment: Analysis



Scheme (I): Complete, Scheme (II): Partial

- When budget is not large enough, scheme II is consistently better than I in all tasks,
- When the budget goes down from 100%, the advantage of II is more prominent;
- but when the budget is too low, the quality of $\tilde{P}$ degrades and hurts the performance, leading to roughly hill-shaped curves.

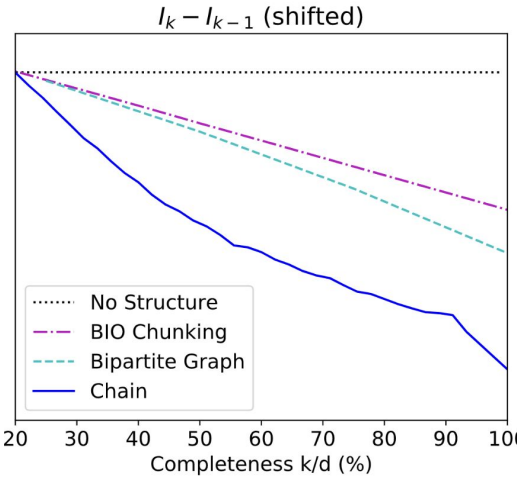
Theory Behind: Main Idea

In General:

Suppose the complete annotation is d-dimension, and partial annotation covers k dimensions.

This paper defines a quantity **(Ik)** to measure the benefit brought by current k-partial annotation.

It further calculates the marginal benefit (Ik - Ik-1) of annotating one more step.



Question: How to get Ik?

Theory Details: Definitions

Define ***Structure:*** A vector of *d* random variables: Y = [Y1 , . . . , Yd] ∈ C(𝔏^d) ⊆ 𝔏^d, where 𝔏 = { ℓ1, ℓ2, … , ℓ|𝔏|} is the label set for each variable and C(𝔏^d) represents the all possible structures under constraints imposed by this type of structure.

➔ E.g. Simple case: When the variables are independent: C(𝔏^d) = 𝔏^d.

Define ***Annotation*****:** A k-step annotation (0 ≤ k ≤ d) is a vector of RVs Ak = [$A_k = [A_{k,1}, A_{k,2}, \dots, A_{k,d}] \in (\mathcal{L} \cup \sqcap)^d$, re ⊓ is a special character for null, s.t.

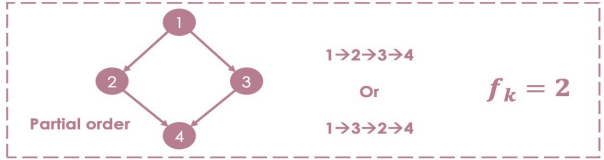
- $\sum_{i=1}^{d} \mathbb{I}(A_{k,i} \neq \sqcap) = k$ ⇒ in total, k variables are annotated at step k.
- $P(Y|A_k = a_k) = P(Y|Y_j = a_{k,j}, j \in \mathcal{J})$, where $\mathcal{J} = \{j : a_{k,j} \neq \sqcap\}$ ⇒ no annotation mistakes
- A_k means k variables in Y are correctly labeled

Define: $\boldsymbol{I_k \triangleq \log |C(\mathcal{L}^d)| - E[\log f_k]}$

- ❑ fk is **the number of all possible structures** given Ak.
- ❑ Ik measures how much of C(𝔏^d) has been **disqualified** by **k** labels
  → the theoretical benefit of Ak.
- ★ When Y is unknown, assume Y follows a uniform distribution over C(𝔏^d).
  Then I(Y; Ak) = Ik.

$f_0 = |C(\mathcal{L}^d)| \geq f_1 \geq f_2 \geq \cdots \geq f_d = 1$

No annotation — Complete annotation



Theory Details: Ik for Chunking, SRC, and Temporal Rel

Chunking:

- No closed-form solution to Ik; use dynamic programming simulations to get f(ak), then $I_k = \log |C(\mathcal{L}^d)| - E\left[\log f(\mathbf{a}_k)\right]$

SRC (Bipartite graph structure):

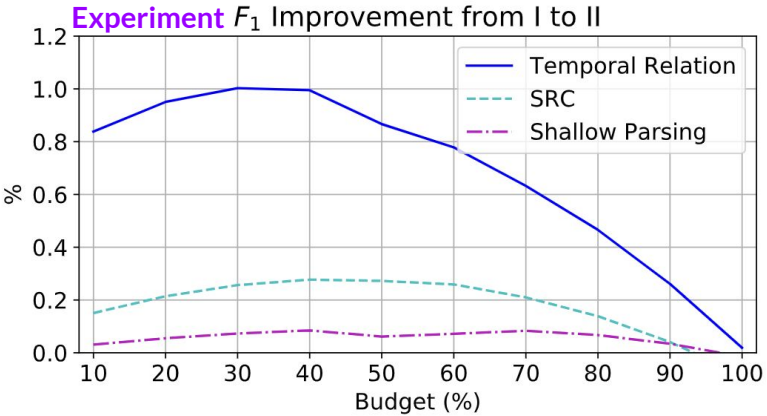
- Assign d agents to d′ tasks. With k out of d agents assigned, we need to assign the remaining (d' − k) tasks to (d − k) agents.
- $I_k = \log \frac{d'!}{(d'-k)!}$

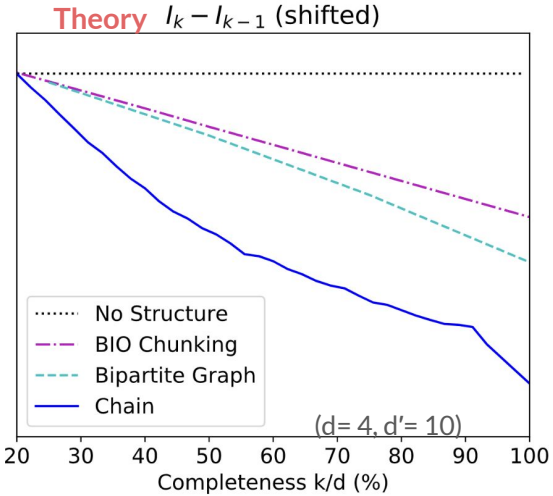
Temporal Relation Extraction (Chain structure in ranking problems):

- When k out of d comparisons are given, the structure is a Directed acyclic graph (DAG), then , f (ak) is actually counting the number of linear extensions of the DAG
- f (ak) is #P complete → use the Kahn's algorithm and backtracking to simulate Ik with a relatively small n.

Theory Proof: Chunking, SRC

Ik − Ik−1 is the benefit brought by annotating an additional variable a





Implications of the curves: **diminishing return of new labels**

The slope may be an indicator for the strengths of structures

- ❏ No structure at all ⇒ the curve is flat.
- ❏ BIO structure is simple ⇒ the flattest slope among 3 tasks.
- ❏ When the structure is a chain, the level of uncertainty goes down rapidly with every single annotation ⇒ the constraint is intuitively strong and it indeed has a steep slope (blue).

Theory corresponds to Experiment Results.

My Comments

Key Contributions: Explains benefits of partial annotation for structures.

My Questions:

1. If the structure data is too complex/new, are there always some general constraints?
2. The cost is uncertain, it is assumed as linear to number of annotations, and same for partial/complete.
3. In the method ESPA, the optimal time to stop is unknown.
4. Besides text, no other data format is tested. What about images, videos, audios?

Poster page: https://www.qiangning.info/papers/NHFR19-poster-final.pdf
Paper page: https://cogcomp.seas.upenn.edu/page/publication_view/868

Thank you!