Partial Or Complete, That's The Question

Qiang Ning[1], Hangfeng He[2], Chuchu Fan[1], Dan Roth[1,2]
[1]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign
[2]Department of Computer and Information Science, University of Pennsylvania

Background

Many learning tasks are **structured**. Considering the structure in the learning/prediction phase leads to structured learning/prediction. **However**, the effect of structure on data collection has received less attention.

Considering structures in different phases

Data Collection	⇢	**This Paper**
↓		
Learning	⇢	Structured Learning
↓		
Prediction	⇢	Structured Prediction

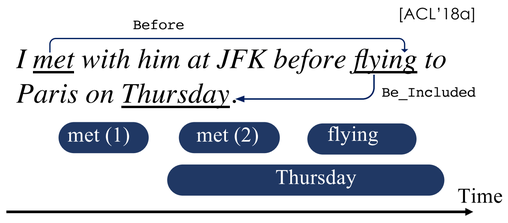
This paper aims to provide a better (both theoretical and empirical) understanding of **the role of structure in data collection for structured tasks**.

Motivation

Structure: a set of variables that are not independent (mathematical definition shown in Sec. "Theory").
A common perception: "partial" data are of low quality and should be avoided in data collection.
This paper **challenges** it and provides a more principled understanding for "partial".

Temporal relation extraction (no time loops)

[ACL'18a]



(met, Thursday) is either "before" or "included"

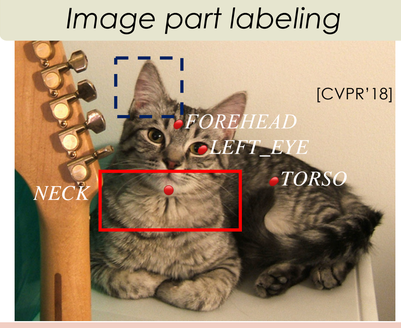
Semantic role labeling (one phrase, one label)

[LREC'02]

"frog" and "girl" cannot be "Arg0" anymore

Image part labeling

[CVPR'18]



"NECK" is in the red box instead of the blue one

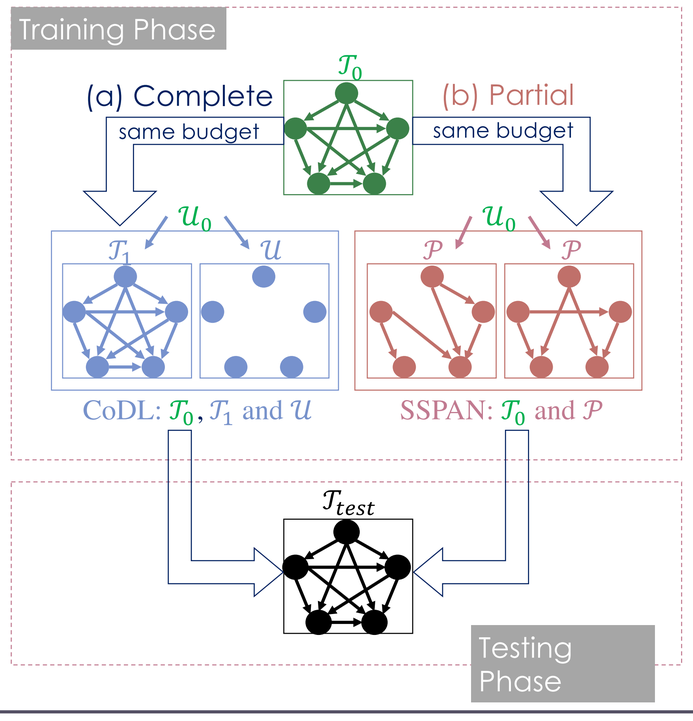
Motivation: if we annotate a structure only partially, the remaining part is constrained to a smaller space; that is, some "information" of the remaining part has been explained by existing annotations. This is a **benefit of "partial" that is overlooked previously** [*SEM'18].

Approach: Partial Or Complete

To study the effect of structure on data collection, we investigate **two data collection paradigms**:

- **Complete**: complete as many structures as possible; the cost is leaving some structures empty
- **Early stopping**: evenly distribute the annotation budget to all structures; the cost is only getting partial annotations

We train two systems accordingly and compare their performances on **the same test set**.



Algorithm: CoDL & SSPAN

CoDL: constraint-driven learning; can be seen as "structured self-learning" [ACL'07]
SSPAN: structured self-learning with partial annotations; can be seen as a straightforward extension of CoDL [*SEM'18]

The SSPAN Algorithm

Input: $\mathcal{T} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}, \mathcal{P} = \{(\boldsymbol{x}_i, \boldsymbol{a}_i)\}_{i=N+1}^{N+M}$
1. Initialize $\mathcal{H} = LEARN(\mathcal{T})$
2. While convergence not reached, do
   1) $\tilde{\mathcal{P}} = \emptyset$
   2) Foreach $(\boldsymbol{x}_i, \boldsymbol{a}_i) \in \mathcal{P}$ do
      i. $\hat{\boldsymbol{y}}_i = INFERENCE(\boldsymbol{x}_i; \mathcal{H})$, s.t.
         $\hat{\boldsymbol{y}}_i \in C(\mathcal{Y}^d)$
         $\hat{y}_{i,j} = a_{i,j}, \forall a_{i,j} \neq \sqcap$ ⭐
      ii. $\tilde{\mathcal{P}} = \tilde{\mathcal{P}} \cup \{\boldsymbol{x}_i, \hat{\boldsymbol{y}}_i\}$
   3) $\mathcal{H} = LEARN(\mathcal{T} + \tilde{\mathcal{P}})$
3. Return $\mathcal{H}$

SSPAN is conceptually a **hard-EM** algorithm for structured learning tasks. Without ⭐, SSPAN goes back to CoDL.

Theory

Structure: a vector of d random variables: $Y = [Y_1, Y_2, \dots, Y_d] \in C(\mathcal{L}^d) \subseteq \mathcal{L}^d$, where $\mathcal{L}$ is the label set.

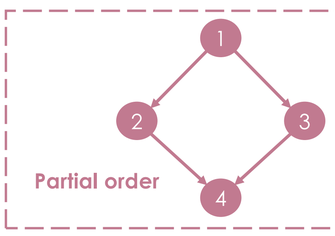
Two simple cases:
- When the variables are independent: $C(\mathcal{L}^d) = \mathcal{L}^d$
- When the structure is so "strong" that it requires all variables to share the same label: $C(\mathcal{L}^d) = \{[\ell_1, \ell_1, \dots, \ell_1], [\ell_2, \ell_2, \dots, \ell_2], \dots, [\ell_{|\mathcal{L}|}, \ell_{|\mathcal{L}|}, \dots, \ell_{|\mathcal{L}|}\}$

Complete annotation: All variables are labeled and a **unique point in $C(\mathcal{L}^d)$ is determined.**

Partial annotation: A subset of $C(\mathcal{L}^d)$ is determined, which contains the "true" structure.

- Let f_k **be the size of the feasible subset** if k out of d variables are labeled.
- $f_0 = |C(\mathcal{L}^d)| \geq f_1 \geq f_2 \geq \cdots \geq f_d = 1$

No annotation — Complete annotation



1→2→3→4
Or
1→3→2→4
$f_k = 2$

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

- Measures how much of $C(\mathcal{L}^d)$ has been **disqualified** by k labels.

Define: A k**-partial annotation** A_k is a vector of random variables $A_k = [A_{k,1}, A_{k,2}, \dots, A_{k,d}] \in (\mathcal{L} \cup \sqcap)^d$, where $\sqcap$ is a special character for no label yet, s.t.

- $\sum_{i=1}^{d} \mathbb{I}(A_{k,i} \neq \sqcap) = k$
- $P(Y|A_k = a_k) = P(Y|Y_j = a_{k,j}, j \in \mathcal{J})$, where $\mathcal{J} = \{j: a_{k,j} \neq \sqcap\}$
- A_k means k variables in Y are correctly labeled

Theorem: I_k is the **mutual information** between Y and A_k when both Y and the k variables labeled in A_k follow uniform distributions.

Chain structure in ranking problems:

- *Linear chain with a transitivity constraint.*
- When k out of d comparisons are given, the structure is a partial order.
- f_k: # of linear extensions of these partial orders; need simulations to estimate I_k [ToC'91].

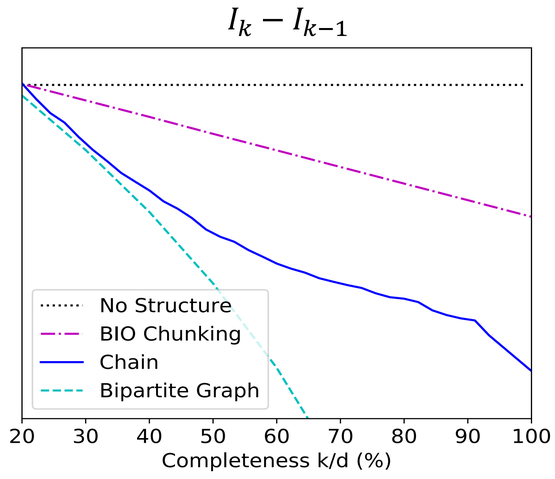
Bipartite graph structure in assignment problems:

- *Assign d agents to d' tasks (w.l.o.g, $d < d'$) such that each agent handles exactly one task, and each task can only be handled by at most one agent.*
- When k out of d agents are assigned, we need to assign the remaining $d' - k$ tasks to $d - k$ agents.
- $I_k = \log \frac{d'!}{(d'-k)!}$

Sequence tagging problems:

- *Shallow Parsing, NER, etc., are key examples.*
- *O cannot be immediately followed by I.*
- No closed-form solution to I_k; need dynamic programming simulations.

$I_k - I_{k-1}$

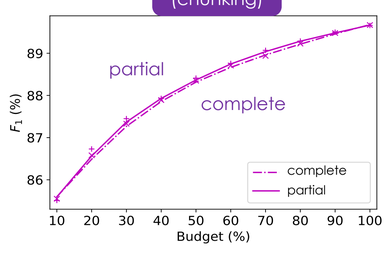


Implications of the curves above:

- Diminishing return of new labels
- Better to annotate a structure partially than completely
- The slope may be an indicator for the strengths of structures

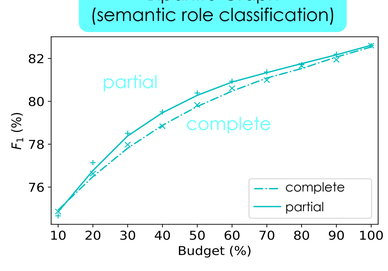
Experiments

Seq Tagging (chunking)



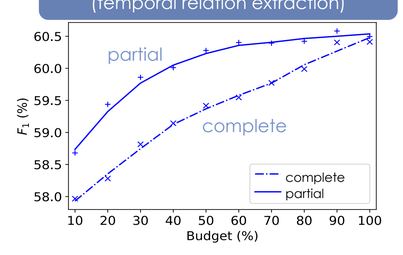
[CoNLL'00, NIPS'01]

Bipartite Graph (semantic role classification)



[CoNLL'05, LREC'18]

Chain (temporal relation extraction)



[ACL'18b, EMNLP'18]

Results:

- Partial > Complete
- The stronger a structure, the more improvement from partial.

Conclusion

This paper provides a **unique view of structured annotations**: it is the reduction in the uncertainty of a target structure Y, by a random process A representing the annotation process.

- **Theoretically**, "partial" provides more information
- **Empirically**, "partial" leads to improved performance on three very different tasks

In general, we argue that any signal that has **non-zero mutual information** with Y can be viewed as "annotation".

References

[ToC'91] G. Brightwell and P. Winkler. Counting linear extensions is #p-complete. **[CoNLL'00]** TK Sang and S. Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. **[NIPS'01]** V. Punyakanok and D. Roth. The use of classifiers in sequential inference. **[LREC'02]** P. Kingsbury and M. Palmer. From Treebank to PropBank. **[CoNLL'05]** X. Carreras and L. Marquez. Introduction to the CoNLL-2005 shared tasks: Semantic role labeling. **[ACL'07]** MW Chang, L. Ratinov, and D. Roth. Guiding semi-supervision with constraint-driven learning. **[ACL'18a]** Q. Ning, Z. Feng, H. Wu, and D. Roth. Joint reasoning for temporal and causal relations. **[ACL'18b]** Q. Ning, H. Wu, and D. Roth. A multi-axis annotation scheme for event temporal relations. **[CVPR'18]** J. Choi, J. Krishnamurthy, A. Kembhavi, and A. Farhadi. Structured set matching networks for one-short part labeling. **[EMNLP'18]** Q. Ning, B. Zhou, Z. Feng, H. Peng, and D. Roth. CogCompTime: A tool for understanding time in natural language. **[LREC'18]** D. Khashabi et al. CogCompNLP: Your swiss army knife for NLP. **[*SEM'18]** Q. Ning, Z. Yu, C. Fan, and D. Roth. Exploiting partially annotated data for temporal relation extraction.

Acknowledgement

The Allen Institute for Artificial Intelligence (allenai.org); the IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network; Contract HR0011-15-2-0025 with the US Defense Advanced Research Projects Agency (DARPA); the Army Research Laboratory (ARL).

ILLINOIS — Cognitive Computation Group — Penn University of Pennsylvania — C3SR center for cognitive computing systems research — IBM | ILLINOIS