Preliminary Report on AKBC For Natural Processes

Dick Chiang[*]
Department of Computer Science
Stony Brook University
Stony Brook, NY 11790
rchiang@cs.stonybrook.edu

Abstract

We envision an automatic knowledge base construction (AKBC) system containing facts about natural processes, and whose primary performance metric shall be its ability to answer grade school science questions. The planned system relies on a semantic role labeller capable of identifying those parts of sentences corresponding to process roles, e.g., "undergoer", "theme." This discriminative capability enables search agents to traverse the internet and collect semantically rich sentences chosen for relevance and role coverage. This short report describes the current state of development and offers some simple technical thoughts for increased collaboration and reproducibility of results.

1 System Synopsis

A semantic role labelling (SRL) system extracts roles from acquired sentences. An integer linear programmer (ILP) then integrates these new relationships into the existing graph, resolving potential contradictions via an objective function maximizing role likelihood and lexical consistency. For example, the noun phrase "the water droplets" may score highest for the role *result* under condensation but may appear in previous sentences as *undergoer*, say for evaporation. The ILP attempts to find the best role in light of these soft global constraints as well as hard local constraints such as the forbidding of repeated roles within a sentence. New extensions to the semantic graph are then translated into Google queries, and the iterative acquisition proceeds.

While the exact details of the actual inferential mechanism remain unclear, a minimal requirement for a question answering system is a semantic graph on which to map candidate queries. To first order, the process represents a verbal predicate. A directed graph is constructed for each such process whose vertices are argument instances or, less formally, roles. For the process *evaporation*, the argument instances could be *perspiration* (undergoer), *wind* (enabler), and *cooling* (result). Thus, regardless of which inferential mechanism we choose—whether classical first-order inference or graphical methods—SRL is of paramount importance.

2 An Argument For Higher Recall

Louvan et al. 2015 reported a best precision/recall of 0.5614/0.3351 for the SRL task on a set of 758 process related sentences. Existing SRL systems appear to perform well on argument classification but poorly on argument identification. That is, given a set of process-specific sentences for which a human annotator has identified arguments fulfilling key roles, the SRL system often does select the same role as the human but only for a minority of spans it does not otherwise label as *None*. The fact that most spans are deemed as not filling any predetermined role results in the subpar recall score. This is a weakness of SRL systems which combine the argument classification and identification tasks.

For the present task, unfortunately, argument identification is more important than argument classification. Balasubramanian 2015 notes that for the purposes of knowledge acquisition, the semantic role labeller need not make fine

[*] Submitting for partial satisfaction of the requirements of CSE523

Feature	Description
EntHeadPOS	NN
EntHeadWord	*sediment*
EntityPOSDepRel	NP, true[*sediment* depends on *turned*]
EntHeadEvtPOS	*sediment*, VBN
PathEntToEvt	VBN↑, VP↑, ROOT↓, S↓, VP↓, NP↓, SBAR ↓, S↓, VP↓, VP↓, PP↓, NP↓
EntHeadEvtHead	*sediment*, *turned*
EntNPAndRelatedToEvt	true[*sediment* is NP and depends on *turned*]
EntPOSEntHeadEvtPOS	NP,*sediment*, VBN
EntPOSEvtPOSDepRel	NP, VBN, true
EntPOSEntParentPOSEvtPOS	NP, *sediment*, VBN
PathEntToAncestor	NP↑, PP↑, VP↑, ROOT↓, S↓, S↓, VP↓, NP↓, SBAR↓, S↓, S↓, VP↓, VP↓, VP↓
PathEntToRoot	NP↑, PP↑, VP↑, VP↑, S↑, SBAR↑, NP↑, VP↑, S↑, ROOT↑, ROOT↓
EntParentPOSEvtPOS	*sediment*, VBN

Table 1: SRL features (Scaria et al. 2013)

distinctions between roles so long as it can pick out high-quality sentences. In other words, it is more important to identify a sentence as containing semantically rich components as opposed to knowing what precisely those semantics are.

In terms of precision and recall, the SRL system scores too high on false negatives, that is, it dismisses too many spans as not being one of *undergoer*, *enabler*, *origin*, *destination*, *location*, *theme*, *time* or *result*. While it has been conjectured that a basic set of four to five such roles may not be expressive enough to capture the process semantics, and that the set of roles should be expanded, the preliminary evidence is to the contrary. In nearly all cases of false negatives, the span in question could readily be classified by a human as belonging to one of the pre-determined roles. An ILP might increase precision but seems unlikely to increase recall since its imposition of global constraints would force more role assignments to be null.

3 A Reproducible System

We ran experiments using the open source BIOPROCESS system of Scaria et al. 2013. The SRL features are all typically lexico-syntactic, which Balasubramanian 2015, Gildea and Jurafsky 2002 have lamented as requiring large amounts of training data to be effective. Table 1 shows a representative example of the feature values. In determining optimal feature weights, the BIOPROCESS software employs the Stanford CoreNLP linear classifier (Manning et al. 2014) which applies Quasi-newton minimization to a log conditional objective function. We ran the system using five-fold cross-validation on a combined corpus of 185 paragraphs from the AI2 ProcessBank data (Berant et al. 2014) and 112 sentences from Louvan et al. 2015. Those 112 sentences were chosen from the 785 available for their annotation of the role *trigger*. The BIOPROCESS system like all Propbank inspired systems relies on the identification of a verbal predicate.

The precision, recall, and F1 for this baseline averaged over the five folds are 0.635, 0.275, and 0.383 respectively. However, we see in Table 2 that performance very much depends on the number of training samples per role. Annotations for *undergoer* and *theme* comprise roughly two-thirds of the annotations. If we limit the analysis to those two roles, we see a marked improvement in the recall score. To further improve recall, we consider reducing the multi-class task into a binary one by lumping all semantic roles into a collective class SEMANTIC and setting it against the alternative NONSEMANTIC. We also consider a binary classifier for each individual role to possibly improve argument identification. Table 3 shows the results using unregularized logistic regression.

Roles	$\frac{\#annotations}{total}$	Precision	Recall	F1
Undergoer	0.37	0.74	0.39	0.51
+ Theme	0.67	0.64	0.37	0.47
+ Result	0.79	0.63	0.33	0.44
+ Location	0.87	0.64	0.31	0.42
+ Enabler	0.93	0.63	0.30	0.40
+ Destination	0.96	0.64	0.29	0.40
+ Time	0.98	0.64	0.28	0.39
+ Origin	1.00	0.64	0.28	0.39

Table 2: Recall deteriorates as we add roles with fewer training labels

Certainly in the preliminary stages of the IE task, we could consider employing the much simplified SEMANTIC classifier for its stronger recall. The distinctions between roles would be of secondary concern when the task is to identify text that expresses multiple semantic roles, whatever they may be.

Role	Precision	Recall	F1
SEMANTIC	0.70	0.48	0.57
Undergoer	0.81	0.35	0.49
Theme	0.73	0.19	0.30
Result	0.67	0.07	0.13
Location	0.90	0.07	0.12
Enabler	1.00	0.00	0.00
Destination	1.00	0.01	0.03
Time	1.00	0.00	0.00
Origin	1.00	0.00	0.00

Table 3: Using a logistic classifier per role

4 Software Considerations

It is difficult to realize or even imagine realizing the end-to-end goal drawn in Figure 2 of Balasubramanian 2015, without establishing a *live*, working stubbed-out system on `ambiguity` no matter how humble it currently is. A public alpha system encourages software transparency between lab personnel and provides a tangible baseline for experimentation and discussion.

The author would be mildly interested in sudo privileges on `ambiguity` to begin fleshing out the envisioned computing environment.

4.1 Java-Python Complex

The current practice of feeding static json files containing role probabilities into the Python ILP software seems fraught with reproduciblity issues. In particular, there could be software errors in the writing of the data, software errors in the reading of the data, and/or miscommunication as to what the fields signify. The files could inadvertently change on disk (so-called bitrot). Any improvements to either the quality of the data or the classifier itself would go unnoticed, or at least would need to wait until a future release, at which point a number of other formatting changes could be introduced to jeopardize correctness. Ideally, we wish to live in a world where the ILP maintainer, as a matter of course in running his experiments, can reproduce the inputs at will and feed the input data directly from memory without the awkward relay over disk. In addition to the logistic benefits, he would gain greater insight into the data and could suggest/make improvements to the SRL core.

To that end, we have written a short script `PickleSRL.py` which runs the Java SRL system and stores the results for consumption by Python while avoiding the dangers of writing and reading json formats. We hope others in the lab will come to adopt this style of tighter code integration. We also encourage Scala for lightweight jobs whenever manipulation of objects within the Java SRL core is necessary.

5 Continued Work

Table 2 convincingly shows more training annotations would improve SRL performance to acceptable levels. We believe development of the AKBC should continue in a supervised manner. The exploration of unsupervised methods, while theoretically interesting, would be difficult to apply to a system still in its infancy. Increased code sharing and a lab-wide infrastructure integrating the Java back-end, the Mongo data store, and the Python data tools would accelerate the pace of development and help to ensure correctness. With a baseline SRL now established, the author would like, as a next step, to clarify how process knowledge is represented in the store. How precisely does SRL-ILP inform the construction and expansion of the semantic graphs mentioned in the introduction? A related and important question is how best to rationally taxonomize natural processes. While these are questions more relevant to the engineering than the science, they are nonetheless crucial for proof of concept.

References

[1] N. Balasubramanian. “CRII:III: Composing Process Knowledge Using Semantic Roles”. unpublished project paper. 2015.

[2] J. Berant et al. *AI2 ProcessBank Dataset*. 2014. URL: `allenai.org/data.html`.

[3] Daniel Gildea and Daniel Jurafsky. “Automatic Labeling of Semantic Roles”. In: *Comput. Linguist.* 28.3 (Sept. 2002), pp. 245–288. ISSN: 0891-2017. DOI: `10.1162/089120102760275983`. URL: `http://dx.doi.org/10.1162/089120102760275983`.

[4] S. Louvan et al. “Semantic Role Labeling for Process Recognition Questions”. In: *K-CAP Scientific Knowledge Workshop*. 2015.

[5] Christopher D. Manning et al. “The Stanford CoreNLP Natural Language Processing Toolkit”. In: *Association for Computational Linguistics (ACL) System Demonstrations*. 2014, pp. 55–60. URL: `http://www.aclweb.org/anthology/P/P14/P14-5010`.

[6] A. Scaria et al. “Learning Biological Processes with Global Constraints”. In: *Proceedings of EMNLP 2013*. 2013.