An Evaluation of PredPatt and Open IE via Stage 1 Semantic Role Labeling

Sheng Zhang
Johns Hopkins University
zsheng2@jhu.edu

Rachel Rudinger
Johns Hopkins University
rudinger@jhu.edu

Benjamin Van Durme
Johns Hopkins University
vandurme@cs.jhu.edu

Abstract

PredPatt is a pattern-based framework for predicate-argument extraction. While it works across languages and provides a well-formed syntax-semantics interface for NLP tasks, a large-scale and reproducible evaluation has been lacking, which prevents comparisons between PredPatt and other related systems, and inhibits the updates of the patterns in PredPatt. In this work, we improve and evaluate PredPatt by introducing a large set of high-quality annotations converted from PropBank, which can also be used as a benchmark for other predicate-argument extraction systems. We compare PredPatt with other prominent systems and shows that PredPatt achieves the best precision and recall.

1 Introduction

PredPatt[1] (White et al., 2016) is a pattern-based framework for predicate-argument extraction. It defines a set of interpretable, extensible and non-lexicalized patterns based on Universal Dependencies (UD) (de Marneffe et al., 2014), and extracts predicates and arguments through these manual patterns. Figure 1 shows the predicates and arguments extracted by PredPatt from the sentence: "*Chris, the designer, wants to launch a new brand.*"

(1) [Chris, the designer] **wants** [to launch a new brand]
(2) [Chris, the designer] **to launch** [a new brand]
(3) [Chris] **be** [the designer]

Figure 1: Predicates and arguments extracted by PredPatt.[2]

The underlying predicate-argument structure constructed by PredPatt is a directed graph, where a special dependency `ARG` is built between a predicate head token and its arguments' head tokens, and the original UD relations are retained within predicate phrases and argument phrases. For example, Figure 2 shows the directed graph for the predicate-argument extraction (1) and (2) in Figure 1.

Compared to other existing systems for predicate-argument extraction (Banko et al., 2007; Fader et al., 2011; Angeli et al., 2015), the use of manual language-agnostic patterns on UD makes PredPatt a well-founded component across languages. Additionally, the underlying structure constructed by PredPatt has been shown to be a well-formed syntax-semantics interface for NLP tasks: Zhang et al. (2016) utilizes PredPatt to extract possibilistic propositions in automatic common-sense inference generation. White et al. (2016) uses PredPatt to help augmenting data with *Universal Decompositional Semantics*. Zhang et al. (2017) adapts PredPatt to data generation for cross-lingual open information extraction.

However, the evaluation of PredPatt has been restricted to manually-checked extractions over a small set of sentences (White et al., 2016), which lacks gold annotations to conduct an objective and reproducible evaluation, and inhibits the updates of patterns in PredPatt.

[1] PredPatt is publicly available at https://github.com/hltcoe/PredPatt
[2] The predicates are colored blue, and the arguments are colored purple with brackets.

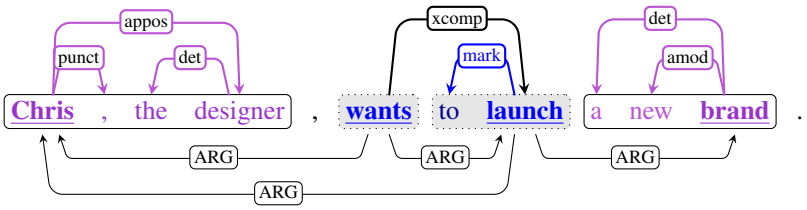

Figure 2: Underlying predicate-argument structure constructed by PredPatt. The predicates are colored blue in dotted cycles with gray background. The arguments are colored purple in solid cycles. The head tokens of predicates and arguments are underlined in bold. A special dependency ARG is built between a predicate head token and its arguments head tokens. The UD relations are kept within predicates and arguments. The relations between predicate head tokens are also kept. The upper relations are UD. The lower relations are ARG relations added by PredPatt.

In this work, we aim to conduct a large-scale and reproducible evaluation of PredPatt by introducing a large set of gold annotations gathered from PropBank (Palmer et al., 2005). We leverage these gold annotations to improve PredPatt and compare it with other prominent systems. The evaluation results demonstrate that we make a promising improvement on PredPatt, and it significantly outperforms other comparing systems. The scripts for creating gold annotations and evaluation are available at: https://github.com/hltcoe/PredPatt/tree/master/eval

2 Creating Gold Annotations

Open Information Extraction (Open IE) and Semantic Role Labeling (SRL) (Carreras and Màrquez, 2005) are quite related: semantically labeled arguments correspond to the arguments in Open IE extractions, and verbs often match up with Open IE relations (Christensen et al., 2011). Lang and Lapata (2010) has acknowledged that the SRL task can be viewed as a two stage process of (1) recognizing predicates and arguments then (2) assigning semantics. Therefore, predicate-argument extraction (i.e., Open IE) should primarily be considered the same as the first of two stages of SRL, and expert annotated SRL data would be an ideal resource for evaluating Open IE systems. This makes PropBank (Palmer et al., 2005) a natural choice from which we can create gold annotations for Open IE, Here, we choose to use expert annotations from PropBank, as compared to the recent suggestion to employ non-expert annotations as a means of benchmarking systems Stanovsky and Dagan (2016). Another advantage of choosing PropBank is that PropBank has gold annotations for UD which lays the important groundwork for evaluating UD-based patterns in PredPatt.

In this work, we create gold annotations for predicate-argument extraction by converting PropBank annotations on English Web Treebank (EWT) (LDC2012T13) and the Penn Treebank II Wall Street Journal Corpus (WSJ) (Marcus et al., 1994).[3] These two corpora have all verbal predicates annotated, and are used to evaluate PredPatt in different perspectives: EWT is the corpus where the gold standard English UD Treebank is built over, which enables an evaluation and analysis of PredPatt patterns; WSJ is used to evaluate PredPatt in a real-world scenario where we run SyntaxNet Parser[4] (Andor et al., 2016) on the corpus to generate automated UD parses as input of PredPatt.

Table 1 shows the statistics of the auto-converted gold annotations for predicate-argument extraction on EWT and WSJ. We convert the PropBank annotations for all verbal predicates in these two corpora, and ignore roles of directional (DIR), manner (MNR), modals (MOD), negation (NEG) and adverbials (ADV), as they aren't extracted as distinct argument but instead are folded into the complex predicate by PredPatt and other systems for predicate-argument extraction (Banko et al., 2007; Fader et al., 2011; Angeli et al., 2015). For EWT, we select 13,583 sentences that have the version 2.0 of the gold UD annotations.[5] The resulting annotations on these two corpora contain over 94K extractions.

[3] PropBank annotations are available at: https://github.com/propbank/propbank-release

[4] SyntaxNet Parser is trained on the UD Treebank which has no overlap with WSJ.

[5] English Universal Depedency Treebank is available at: http://universaldependencies.org

Corpus	#sentence	#predicate	#unique_verb	#avg_arg_per_pred
EWT	13,583	21,479	4,336	2.0
WSJ	36,432	73,076	7,880	2.1

Table 1: Statistics of the gold annotations on EWT and WSJ.

3 Improving PredPatt

PredPatt is a pattern-based system, comprising an extensible set of clean, interpretable linguistic patterns over UD parses. By analyzing PredPatt extractions in comparison with gold annotations (Sec. 2), we are able to refine and improve PredPatt's pattern set. From the auto-converted gold annotations, we create a held-out set by randomly sampling 10% sentences from EWT. We then update the existing PredPatt patterns and introduce new patterns by analyzing PredPatt annotations on the held-out set.

PredPatt extracts predicates and arguments in four stages (White et al., 2016): (1) predicate and argument root identification, (2) argument resolution, (3) predicate and argument phrase extraction, and (4) optional post-processing. We analyze PredPatt extraction in each of these stages on the held-out set, and make 19 improvements to PredPatt patterns. Due to lack of space, we only highlight one improvement for each stage below.

Fixed-MWE-pred: The UD version 2.0 introduces a new dependency relation `fixed` for identifying fixed function-word "multiword expressions" (MWEs). To accommodate this new feature, we add patterns to identify the MWE predicate and its argument. As shown in Figure 3, the predicate root in this case is the dependent of `fixed` that is tagged as a verb (i.e., "opposed"); the root of its argument is the token which indirectly governs the predicate root via the `case` and `fixed` relation (i.e., "one").

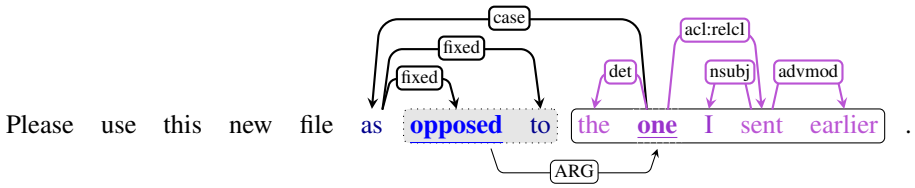

Figure 3: Example for add argument for `fixed` MWE predicates.

Cut-complex-pred: The existing patterns take clausal complements (`ccomp` and `xcomp`) as *predicatives* of complex predicates in the argument resolution stage, where the arguments of the clausal complement will be merged into the argument set of their head predicate. For example, in the sentence "*Chris, the designer, wants to launch a new brand*", PredPatt merges the argument "*a new brand*" of the predicate "*to launch*" into the argument set of the complex predicate "*wants to launch*". As a result, only the complex predicate, "[Chris, the designer] **wants to launch** [a new brand]", will be extracted. It ignores the possibility of the clausal complement itself being a predicate. Here, we add a cutting option; when turned on, it will cut the complex predicate into simple predicates as shown in Figure 1.

Prep-separation: By default, PredPatt considers prepositions to belong to the predicate, while PropBank places preopositions within the span of their corresponding argument. Either behavior may be preferable under different circumstances, so we make preposition placement a new configurable option of PredPatt.

Borrow-subj-for-conj-of-xcomp: PredPatt contains a post-processing option for distributing a single `nsubj` argument over multiple predicates joined by a `conj` relation. PredPatt also contains a pattern assigning subject arguments to predicates introduced by open clausal complement (`xcomp`) relations, according to the theory of obligatory control (Farkas, 1988). We introduce a new post-processing option that combines these two patterns, allowing an argument in subject position to be distributed over multiple `xcomp` predicates that are joined by a `conj` relation, as illustrated in Figure 4.

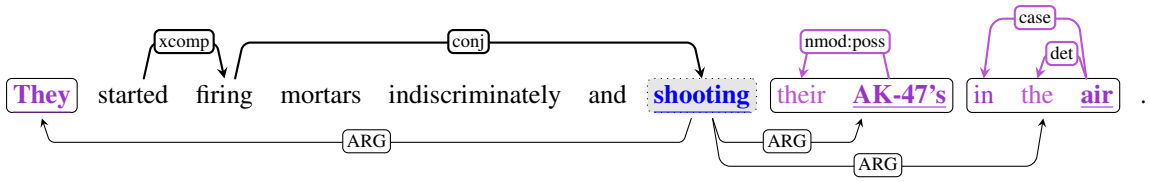

Figure 4: Example for borrowing subject from the conjunction of open clausal complement.

4 Evaluation

In this section, we evaluate the original PredPatt (PredPatt v1) and the improved PredPatt (PredPatt v2) on the English Web Treebank (EWT) and the Wall Street Journal corpus (WSJ), and compare their performance with four prominent Open IE systems: OpenIE 4,[6] OLLIE (Mausam et al., 2012), ClausIE (Del Corro and Gemulla, 2013), and Stanford Open IE (Angeli et al., 2015).

4.1 Precision-Recall Curve

We compare PredPatt with four prominent Open IE systems which are also built for predicate-argument extraction. To allow some flexibility, we compute the precision and recall of different systems by running the scripts used in Stanovsky and Dagan (2016),[7] where an automated extraction is matched with a gold extraction based on their token-level overlap. Figure 5 and Figure 6 show the Precision-Recall Curves for different systems on EWT and WSJ.[8] When tested on EWT which has gold UD parses (Figure 5), PredPatt v1 and v2 outperforms the other systems by a significant margin in both precision and recall. When tested on WSJ where only automated UD parses are available (Figure 6), ClausIE achieves a recall that is slightly better than PredPatt v1, but PredPatt v2 still shows the best performance across all systems.

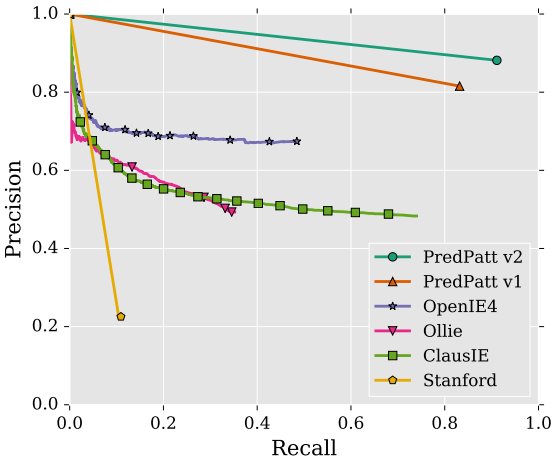

Figure 5: Precision-Recall Curve for different systems on EWT w/ gold UD.

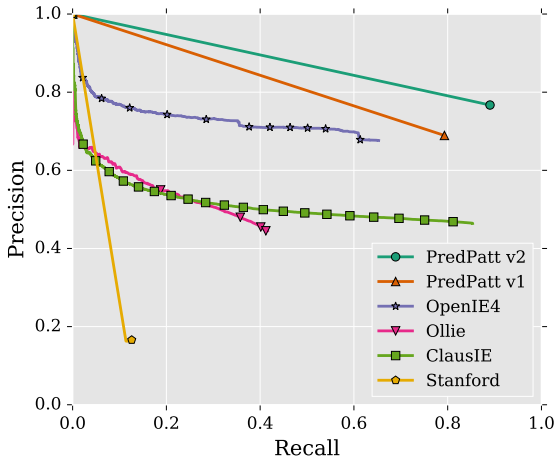

Figure 6: Precision-Recall Curve for different systems on WSJ w/ automated UD.

4.2 Extraction Head Agreement

The rich underlying structure in PredPatt (see Figure 2) contains head information for predicates and arguments, which enables a precision-recall metric based on the agreement of head information. Similar

[6] OpenIE 4 is available at: https://github.com/allenai/openie-standalone.

[7] The scripts are available at: https://github.com/gabrielStanovsky/oie-benchmark.

[8] Studies of PredPatt confidence prediction have been done before, but the current system does not output them. In this evaluation, we assign 1.0 confidence score to all PredPatt extractions.

to He et al. (2015), we first match an automated predicate with a gold predicate if they both agree on their head.[9] With two matched predicates, we then match an automated argument with a gold argument if the automated argument head is within the gold argument span.

We evaluate the precision and recall by a loose macro measure: For the i-th extractions that have two matched predicates, let the argument set of the gold predicate be A_i, and the argument set of the automated predicate be $\hat{A}_i$. The number of matched arguments is represented by $|A_i \cap \hat{A}_i|$. Then the precision is computed by Precision $= \frac{1}{N}\sum_{i=1}^{N} |A_i \cap \hat{A}_i|/|\hat{A}_i|$, and the recall is computed by Recall $= \frac{1}{N}\sum_{i=1}^{N} |A_i \cap \hat{A}_i|/|A_i|$. Table 2 shows the evaluation results of PredPatt v1 and v2 on EWT and WSJ. PredPatt v2 modestly increases the precision by 2.3 on EWT and 0.9 on WSJ, and increases the recall by 1.6 on EWT and 0.2 on WSJ.

	EWT		**WSJ**	
	PredPatt v1	PredPatt v2	PredPatt v1	PredPatt v2
Precision	77.5	79.8 (**+2.3**)	62.1	63.0 (**+0.9**)
Recall	88.0	89.6 (**+1.6**)	84.9	85.1 (**+0.2**)

Table 2: Precision and Recall based on the agreement of head information.

4.3 Statistics of Argument Span Relations

Besides the precion-recall oriented metrics, we impose another metric to further measure the argument span relations. Following in same notations in § 4.2, for the i-th extractions that have an automated predicate and a gold predicate matched with each other, let an argument in the gold argument set be $\alpha \in A_i$, and an argument in the automated argument set $\beta \in \hat{A}_i$. We categorize the automated extractions into four sets according to their arguments relation to the gold arguments.

$$
\begin{aligned}
S_{\text{same}} &= \{(A_i, \hat{A}_i) \mid \forall\alpha \in A_i.\exists\beta \in \hat{A}_i.\text{span}(\alpha) = \text{span}(\beta)\} \\
S_{\text{superset}} &= \{(A_i, \hat{A}_i) \mid \forall\alpha \in A_i.\exists\beta \in \hat{A}_i.\text{span}(\alpha) \subseteq \text{span}(\beta)\} \setminus S_{\text{same}} \\
S_{\text{subset}} &= \{(A_i, \hat{A}_i) \mid \forall\alpha \in A_i.\exists\beta \in \hat{A}_i.\text{span}(\alpha) \supseteq \text{span}(\beta)\} \setminus S_{\text{same}} \\
S_{\text{overlap}} &= \{(A_i, \hat{A}_i) \mid \forall\alpha \in A_i.\exists\beta \in \hat{A}_i.\text{span}(\alpha) \cap \text{span}(\beta) \neq \emptyset\} \setminus (S_{\text{same}} \cup S_{\text{superset}} \cup S_{\text{subset}})
\end{aligned}
$$

Table 3 shows the proportion of PredPatt extractions in different sets. As we expected, compared to WSJ, more extractions on EWT fall into S_{same}, which shows that PredPatt works better on gold UD parses. In contrast to PredPatt v1, PredPatt v2 on EWT increases extractions in S_{same} by 12.97%, which contributes to the most increase of S_{subset}; on WSJ, PredPatt v2 decreases extractions in S_{subset} by 13.89%, which leads the major increases of S_{same} and S_{superset}. There are still over 10% extractions not belonging to any of these four sets. Case analysis shows that the inconsistent extractions are mainly caused by incorrect borrowing of arguments for compound predicates or predicates under obligatory control, missing arguments for passive/active verbs that act as adjectival modifiers, etc. These cases are not easily reachable via UD analysis, but leave room for further improvement on PredPatt.

	EWT		**WSJ**	
	PredPatt v1	PredPatt v2	PredPatt v1	PredPatt v2
Same	63.77	76.74 (**+12.97**)	41.56	52.03 (**+10.47**)
Superset	2.74	4.15 (**+1.41**)	8.64	14.10 (**+5.46**)
Subset	18.31	5.82 (**-12.49**)	28.63	14.74 (**-13.89**)
Overlap	0.78	0.39 (**-0.39**)	2.16	1.06 (**-1.10**)
Other	14.40	12.90 (**-1.50**)	19.01	18.07 (**-0.94**)

Table 3: Proportion of PredPatt extractions in different sets.

[9] In the current settings, the head of a gold predicate is the verb token in the predicate.

5 Conclusions

We introduce a large-scale benchmark for predicate-argument extraction by converting manual annotations from PropBank. Based on the benchmark, we improve PredPatt patterns, and compare PredPatt with four prominent Open IE systems. The comparison shows that PredPatt significantly outperforms the other systems. The evaluation results demonstrate that we improve the performance of PredPatt in both precion-recall and the argument span relation with the gold annotations. As for further work, we see the confidence score estimater for PredPatt extractions as a desirable target, so that the quality of extractions can be controlled. Additionally, we would like to further improve the PredPatt patterns by analyzing more PredPatt extractions in comparison with gold annotations.

Acknowledgments

Thank you to the anonymous reviewers for their feedback, as well as the colleague Tim Vieira. This work was supported in part by the JHU Human Language Technology Center of Excellence (HLTCOE), DARPA LORELEI, and the National Science Foundation. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

References

Andor, D., C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins (2016, August). Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, pp. 2442–2452. Association for Computational Linguistics.

Angeli, G., M. J. Johnson Premkumar, and C. D. Manning (2015, July). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China, pp. 344–354. Association for Computational Linguistics.

Banko, M., M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni (2007). Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, San Francisco, CA, USA, pp. 2670–2676. Morgan Kaufmann Publishers Inc.

Carreras, X. and L. Màrquez (2005). Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pp. 152–164. Association for Computational Linguistics.

Christensen, J., S. Soderland, O. Etzioni, et al. (2011). An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pp. 113–120. ACM.

de Marneffe, M.-C., T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, and C. D. Manning (2014). Universal stanford dependencies: A cross-linguistic typology. In *LREC*, Volume 14, pp. 4585–4592.

Del Corro, L. and R. Gemulla (2013). Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 355–366. ACM.

Fader, A., S. Soderland, and O. Etzioni (2011, July). Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., pp. 1535–1545. Association for Computational Linguistics.

Farkas, D. F. (1988). On obligatory control. *Linguistics and Philosophy 11*(1), 27–58.

He, L., M. Lewis, and L. Zettlemoyer (2015, September). Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 643–653. Association for Computational Linguistics.

Lang, J. and M. Lapata (2010, June). Unsupervised induction of semantic roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California, pp. 939–947. Association for Computational Linguistics.

Marcus, M., G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger (1994). The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, Stroudsburg, PA, USA, pp. 114–119. Association for Computational Linguistics.

Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni (2012). Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

Palmer, M., D. Gildea, and P. Kingsbury (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics 31*(1), 71–106.

Stanovsky, G. and I. Dagan (2016, November). Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 2300–2305. Association for Computational Linguistics.

White, A. S., D. Reisinger, K. Sakaguchi, T. Vieira, S. Zhang, R. Rudinger, K. Rawlins, and B. Van Durme (2016, November). Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 1713–1723. Association for Computational Linguistics.

Zhang, S., K. Duh, and B. Van Durme (2017, April). Mt/ie: Cross-lingual open information extraction with neural sequence-to-sequence models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain, pp. 64–70. Association for Computational Linguistics.

Zhang, S., R. Rudinger, K. Duh, and B. Van Durme (2016). Ordinal common-sense inference. *arXiv preprint arXiv:1611.00601*.