Semantic Role Labeling for Open Information Extraction

Janara Christensen, Mausam, Stephen Soderland and Oren Etzioni
University of Washington, Seattle

Abstract

Open Information Extraction is a recent paradigm for machine reading from arbitrary text. In contrast to existing techniques, which have used only shallow syntactic features, we investigate the use of semantic features (semantic roles) for the task of Open IE. We compare TEXTRUNNER (Banko et al., 2007), a state of the art open extractor, with our novel extractor SRL-IE, which is based on UIUC's SRL system (Punyakanok et al., 2008). We find that SRL-IE is robust to noisy heterogeneous Web data and outperforms TEXTRUNNER on extraction quality. On the other hand, TEXTRUNNER performs over 2 orders of magnitude faster and achieves good precision in high locality and high redundancy extractions. These observations enable the construction of hybrid extractors that output higher quality results than TEXTRUNNER and similar quality as SRL-IE in much less time.

1 Introduction

The grand challenge of Machine Reading (Etzioni et al., 2006) requires, as a key step, a scalable system for extracting information from large, heterogeneous, unstructured text. The traditional approaches to information extraction (*e.g.*, (Soderland, 1999; Agichtein and Gravano, 2000)) do not operate at these scales, since they focus attention on a well-defined small set of relations and require large amounts of training data for each relation. The recent *Open Information Extraction* paradigm (Banko et al., 2007) attempts to overcome the knowledge acquisition bottleneck with its relation-independent nature and no manually annotated training data.

We are interested in the best possible technique for Open IE. The TEXTRUNNER Open IE system (Banko and Etzioni, 2008) employs only shallow syntactic features in the extraction process. Avoiding the expensive processing of deep syntactic analysis allowed TEXTRUNNER to process at Web scale. In this paper, we explore the benefits of semantic features and in particular, evaluate the application of semantic role labeling (SRL) to Open IE.

SRL is a popular NLP task that has seen significant progress over the last few years. The advent of hand-constructed semantic resources such as Propbank and Framenet (Martha and Palmer, 2002; Baker et al., 1998) have resulted in semantic role labelers achieving high in-domain precisions.

Our first observation is that semantically labeled arguments in a sentence almost always correspond to the arguments in Open IE extractions. Similarly, the verbs often match up with Open IE relations. These observations lead us to construct a new Open IE extractor based on SRL. We use UIUC's publicly available SRL system (Punyakanok et al., 2008) that is known to be competitive with the state of the art and construct a novel Open IE extractor based on it called SRL-IE.

We first need to evaluate SRL-IE's effectiveness in the context of large scale and heterogeneous input data as found on the Web: because SRL uses deeper analysis we expect SRL-IE to be much slower. Second, SRL is trained on news corpora using a resource like Propbank, and so may face recall loss due to out of vocabulary verbs and precision loss due to different writing styles found on the Web.

In this paper we address several empirical questions. Can SRL-IE, our SRL based extractor, achieve adequate precision/recall on the heterogeneous Web text? What factors influence the relative performance of SRL-IE *vs.* that of TEXTRUNNER (*e.g.*, n-ary *vs.* binary extractions, redundancy, locality, sentence length, out of vocabulary verbs, *etc.*)? In terms of performance, what are the relative tradeoffs between the two? Finally, is it possible to design a hybrid between the two systems to get the best of both the worlds? Our results show that:

1. SRL-IE is surprisingly robust to noisy heterogeneous data and achieves high precision and recall on the Open IE task on Web text.
2. SRL-IE outperforms TEXTRUNNER along dimensions such as recall and precision on complex extractions (*e.g.*, n-ary relations).
3. TEXTRUNNER is over 2 orders of magnitude faster, and achieves good precision for extractions with high system confidence or high locality or when the same fact is extracted from multiple sentences.
4. Hybrid extractors that use a combination of SRL-IE and TEXTRUNNER get the best of both worlds. Our hybrid extractors make effective use of available time and achieve a superior balance of precision-recall, better precision compared to TEXTRUNNER, and better recall compared to both TEXTRUNNER and SRL-IE.

2 Background

Open Information Extraction: The recently popular Open IE (Banko et al., 2007) is an extraction paradigm where the system makes a single data-driven pass over its corpus and extracts a large set of relational tuples without requiring any human input. These tuples attempt to capture the salient relationships expressed in each sentence. For instance, for the sentence, *"McCain fought hard against Obama, but finally lost the election"* an Open IE system would extract two tuples <McCain, fought (hard) against, Obama>, and <McCain, lost, the election>. These tuples can be binary or n-ary, where the relationship is expressed between more than 2 entities such as <Gates Foundation, invested (arg) in, 1 billion dollars, high schools>.

TEXTRUNNER is a state-of-the-art Open IE system that performs extraction in three key steps. (1) A self-supervised learner that outputs a CRF based classifier (that uses unlexicalized features) for extracting relationships. The self-supervised nature alleviates the need for hand-labeled training data and unlexicalized features help scale to the multitudes of relations found on the Web. (2) A single pass extractor that uses shallow syntactic techniques like part of speech tagging, noun phrase chunking and then applies the CRF extractor to extract relationships expressed in natural language sentences. The use of shallow features makes TEXTRUNNER highly efficient. (3) A redundancy based assessor that re-ranks these extractions based on a probabilistic model of redundancy in text (Downey et al., 2005). This exploits the redundancy of information in Web text and assigns higher confidence to extractions occurring multiple times. All these components enable TEXTRUNNER to be a high performance, general, and high quality extractor for heterogeneous Web text.

Semantic Role Labeling: SRL is a common NLP task that consists of detecting semantic arguments associated with a verb in a sentence and their classification into different roles (such as Agent, Patient, Instrument, *etc.*). Given the sentence *"The pearls I left to my son are fake"* an SRL system would conclude that for the verb 'leave', 'I' is the agent, 'pearls' is the patient and 'son' is the benefactor. Because not all roles feature in each verb the roles are commonly divided into meta-roles (A0-A7) and additional common classes such as location, time, *etc.* Each Ai can represent a different role based on the verb, though A0 and A1 most often refer to agents and patients respectively. Availability of lexical resources such as Propbank (Martha and Palmer, 2002), which annotates text with meta-roles for each argument, has enabled significant progress in SRL systems over the last few years.

Recently, there have been many advances in SRL (Toutanova et al., 2008; Johansson and Nugues, 2008; Coppola et al., 2009; Moschitti et al., 2008). We use UIUC-SRL as our base SRL system (Punyakanok et al., 2008). Our choice of the system is guided by the fact that its code is freely available and it is competitive with state of the art (it achieved the highest F1 score on the CoNLL-2005 shared task).

UIUC-SRL operates in four key steps: pruning, argument identification, argument classification and inference. Pruning involves using a full parse tree and heuristic rules to eliminate constituents that are unlikely to be arguments. Argument identification uses a classifier to identify constituents that are potential arguments. In argument classification, another classifier is used, this time to assign role labels to the candidates identified in the previous stage. Argument information is not incorporated across arguments until the inference stage, which uses an integer linear program to make global role predictions.

3 SRL-IE

Our key insight is that semantically labeled arguments in a sentence almost always correspond to the arguments in Open IE extractions. Thus, we can convert the output of UIUC-SRL into an Open IE extraction. We illustrate this conversion process via an example.

Given the sentence, *"Eli Whitney created the cotton gin in 1793,"* TEXTRUNNER extracts two tuples, one binary and one n-ary, as follows:

binary tuple:	arg0	Eli Whitney
	rel	created
	arg1	the cotton gin

n-ary tuple:	arg0	Eli Whitney
	rel	created (arg) in
	arg1	the cotton gin
	arg2	1793

UIUC-SRL labels constituents of a sentence with the role they play in regards to the verb in the sentence. UIUC-SRL will extract:

A0	Eli Whitney
verb	created
A1	the cotton gin
temporal	in 1793

To convert UIUC-SRL output to Open IE format, SRL-IE treats the verb (along with its modifiers and negation, if present) as the relation. Moreover, it assumes SRL's role-labeled arguments as the Open IE arguments related to the relation. The arguments here consist of all entities labeled Ai, as well as any entities that are marked *Direction*, *Location*, or *Temporal*. We order the arguments in the same order as they are in the sentence and with regard to the relation (except for direction, location and temporal, which cannot be arg0 of an Open IE extraction and are placed at the end of argument list). As we are interested in relations, we consider only extractions that have at least two arguments.

In doing this conversion, we naturally ignore part of the semantic information (such as distinctions between various Ai's) that UIUC-SRL provides. In this conversion process an SRL extraction that was correct in the original format will never be changed to an incorrect Open IE extraction. However, an incorrectly labeled SRL extraction could still convert to a correct Open IE extraction, if the arguments were correctly identified but incorrectly labeled.

Because of the methodology that TEXTRUNNER uses to extract relations, for n-ary extractions of the form <arg0, rel, arg1, ..., argN>, TEXTRUNNER often extracts sub-parts <arg0, rel, arg1>, <arg0, rel, arg1, arg2>, ..., <arg0, rel, arg1, ..., argN-1>. UIUC-SRL, however, extracts at most only one relation for each verb in the sentence. For a fair comparison, we create additional subpart extractions for each UIUC-SRL extraction using a similar policy.

4 Qualitative Comparison of Extractors

In order to understand SRL-IE better, we first compare with TEXTRUNNER in a variety of scenarios, such as sentences with lists, complex sentences, sentences with out of vocabulary verbs, *etc.*

Argument boundaries: SRL-IE is lenient in deciding what constitutes an argument and tends to err on the side of including too much rather than too little; TEXTRUNNER is much more conservative, sometimes to the extent of omitting crucial information, particularly post-modifying clauses and PPs. For example, TEXTRUNNER extracts <Bunsen, invented, a device> from the sentence *"Bunsen invented a device called the Spectroscope"*. SRL-IE includes the entire phrase "a device called the Spectroscope" as the second argument. Generally, the longer arguments in SRL-IE are more informative than TEXTRUNNER's succinct ones. On the other hand, TEXTRUNNER's arguments normalize better leading to an effective use of redundancy in ranking.

Lists: In sentences with a comma-separated lists of nouns, SRL-IE creates one extraction and treats the entire list as the argument, whereas TEXTRUNNER separates them into several relations, one for each item in the list.

Out of vocabulary verbs: While we expected

TEXTRUNNER to handle unknown verbs with little difficulty due to its unlexicalized nature, SRL-IE could have had severe trouble leading to a limited applicability in the context of Web text. However, contrary to our expectations, UIUC-SRL has a graceful policy to handle new verbs by attempting to identify A0 (the agent) and A1 (the patient) and leaving out the higher numbered ones. In practice, this is very effective – SRL-IE recognizes the verb and its two arguments correctly in *"Larry Page googled his name and launched a new revolution."*

Part-of-speech ambiguity: Both SRL-IE and TEXTRUNNER have difficulty when noun phrases have an identical spelling with a verb. For example, the word 'write' when used as a noun causes trouble for both systems. In the sentence, *"Be sure the file has write permission."* SRL-IE and TEXTRUNNER both extract <the file, write, permission>.

Complex sentences: Because TEXTRUNNER only uses shallow syntactic features it has a harder time on sentences with complex structure. SRL-IE, because of its deeper processing, can better handle complex syntax and long-range dependencies, although occasionally complex sentences will create parsing errors causing difficulties for SRL-IE.

N-ary relations: Both extractors suffer significant quality loss in n-ary extractions compared to binary. A key problem is prepositional phrase attachment, deciding whether the phrase associates with arg1 or with the verb.

5 Experimental Results

In our quantitative evaluation we attempt to answer two key questions: (1) what is the relative difference in performance of SRL-IE and TEXTRUNNER on precision, recall and computation time? And, (2) what factors influence the relative performance of the two systems? We explore the first question in Section 5.2 and the second in Section 5.3.

5.1 Dataset

Our goal is to explore the behavior of TEXTRUNNER and SRL-IE on a large scale dataset containing redundant information, since redundancy has been shown to immensely benefit Web-based Open IE extractors. At the same time, the test set must be a manageable size, due to SRL-IE's relatively slow processing time. We constructed a test set that approximates Web-scale distribution of extractions for five target relations – *invent, graduate, study, write,* and *develop*.

We created our test set as follows. We queried a corpus of 500M Web documents for a sample of sentences with these verbs (or their inflected forms, *e.g.*, invents, invented, *etc.*). We then ran TEXTRUNNER and SRL-IE on those sentences to find 200 distinct values of arg0 for each target relation, 100 from each system. We searched for at most 100 sentences that contain both the verb-form and arg0. This resulted in a test set of an average of 6,000 sentences per relation, for a total of 29,842 sentences. We use this test set for all experiments in this paper.

In order to compute precision and recall on this dataset, we tagged extractions by TEXTRUNNER and by SRL-IE as correct or errors. A tuple is correct if the arguments have correct boundaries and the relation accurately expresses the relationship between all of the arguments. Our definition of correct boundaries does not favor either system over the other. For instance, while TEXTRUNNER extracts <Bunsen, invented, a device> from the sentence *"Bunsen invented a device called the Spectroscope"*, and SRL-IE includes the entire phrase "a device called the Spectroscope" as the second argument, both extractions would be marked as correct.

Determining the absolute recall in these experiments is precluded by the amount of hand labeling necessary and the ambiguity of such a task. Instead, we compute pseudo-recall by taking the union of correct tuples from both methods as denominator.[1]

5.2 Relative Performance

Table 1 shows the performance of TEXTRUNNER and SRL-IE on this dataset. Since TEXTRUNNER can output different points on the precision-recall curve based on the confidence of the CRF we choose the point that maximizes F1.

SRL-IE achieved much higher recall at substantially higher precision. This was, however, at the cost of a much larger processing time. For our dataset, TEXTRUNNER took 6.3 minutes and SRL-

[1] Tuples from the two systems are considered equivalent if for the relation and each argument, the extracted phrases are equal or if one phrase is contained within the phrase extracted by the other system.

	TextRunner			SRL-IE		
	P	**R**	**F1**	**P**	**R**	**F1**
Binary	51.9	27.2	35.7	64.4	85.9	73.7
N-ary	39.3	28.2	32.9	54.4	62.7	58.3
All	47.9	27.5	34.9	62.1	79.9	69.9
Time	6.3 minutes			52.1 hours		

Table 1: SRL-IE outperforms TextRunner in both recall and precision, but has over 2.5 orders of magnitude longer run time.

IE took 52.1 hours – roughly 2.5 orders of magnitude longer. We ran our experiments on quad-core 2.8GHz processors with 4GB of memory.

It is important to note that our results for TextRunner are different from prior results (Banko, 2009). This is primarily due to a few operational criteria (such as focusing on proper nouns, filtering relatively infrequent extractions) identified in prior work that resulted in much higher precision, probably at significant cost of recall.

5.3 Comparison under Different Conditions

Although SRL-IE has higher overall precision, there are some conditions under which TextRunner has superior precision. We analyze the performance of these two systems along three key dimensions: system confidence, redundancy, and locality.

System Confidence: TextRunner's CRF-based extractor outputs a confidence score which can be varied to explore different points in the precision-recall space. Figure 1(a) and Figure 2(a) report the results from ranking extractions by this confidence value. For both binary and n-ary extractions the confidence value improves TextRunner's precision and for binary the high precision end has approximately the same precision as SRL-IE. Because of its use of an integer linear program, SRL-IE does not associate confidence values with extractions and is shown as a point in these figures.

Redundancy: In this experiment we use the redundancy of extractions as a measure of confidence. Here redundancy is the number of times a relation has been extracted from unique sentences. We compute redundancy over normalized extractions, ignoring noun modifiers, adverbs, and verb inflection.

Figure 1(b) and Figure 2(b) display the results for binary and n-ary extractions, ranked by redundancy. We use a log scale on the x-axis since high redundancy extractions account for less than 1% of the recall. For binary extractions, redundancy improved TextRunner's precision significantly, but at a dramatic loss in recall. TextRunner achieved 0.8 precision with 0.001 recall at redundancy of 10 and higher. For highly redundant information (common concepts, *etc.*) TextRunner has higher precision than SRL-IE and would be the algorithm of choice.

In n-ary relations for TextRunner and in binary relations for SRL-IE, redundancy actually hurts precision. These extractions tend to be so specific that genuine redundancy is rare, and the highest frequency extractions are often systematic errors. For example, the most frequent SRL-IE extraction was <nothing, write, home>.

Locality: Our experiments with TextRunner led us to discover a new validation scheme for the extractions – *locality*. We observed that TextRunner's shallow features can identify relations more reliably when the arguments are closer to each other in the sentence. Figure 1(c) and Figure 2(c) report the results from ranking extractions by the number of tokens that separate the first and last arguments.

We find a clear correlation between locality and precision of TextRunner, with precision 0.77 at recall 0.18 for TextRunner where the distance is 4 tokens or less for binary extractions. For n-ary relations, TextRunner can match SRL-IE's precision of 0.54 at recall 0.13. SRL-IE remains largely unaffected by locality, probably due to the parsing used in SRL.

6 A TextRunner SRL-IE Hybrid

We now present two hybrid systems that combine the strengths of TextRunner (fast processing time and high precision on a subset of sentences) with the strengths of SRL-IE (higher recall and better handling of long-range dependencies). This is set in a scenario where we have a limited budget on computational time and we need a high performance extractor that utilizes the available time efficiently.

Our approach is to run TextRunner on all sentences, and then determine the order in which to process sentences with SRL-IE. We can increase precision by filtering out TextRunner extractions that are expected to have low precision.

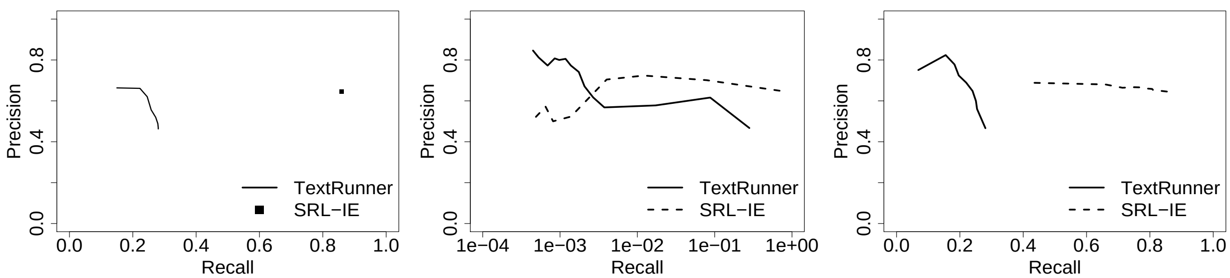

Figure 1: Ranking mechanisms for binary relations. (a) The confidence specified by the CRF improves TEXTRUNNER's precision. (b) For extractions with highest redundancy, TEXTRUNNER has higher precision than SRL-IE. Note the log scale for the x-axis. (c) Ranking by the distance between arguments gives a large boost to TEXTRUNNER's precision.

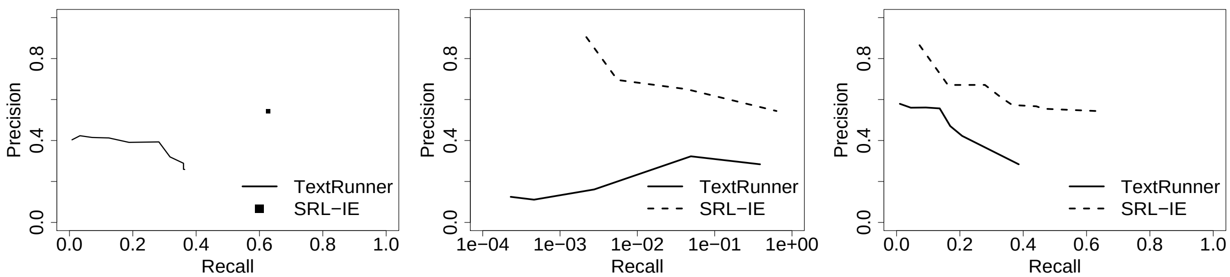

Figure 2: Ranking mechanisms for n-ary relations. (a) Ranking by confidence gives a slight boost to TEXTRUNNER's precision. (b) Redundancy helps SRL-IE, but not TEXTRUNNER. Note the log scale for the x-axis. (c) Ranking by distance between arguments raises precision for TEXTRUNNER and SRL-IE.

A naive hybrid will run TEXTRUNNER over all the sentences and use the remaining time to run SRL-IE on a random subset of the sentences and take the union of all extractions. We refer to this version as RECALLHYBRID, since this does not lose any extractions, achieving highest possible recall.

A second hybrid, which we call PRECHYBRID, focuses on increasing the precision and uses the *filter policy* and an intelligent *order of sentences for extraction* as described below.

Filter Policy for TEXTRUNNER Extractions: The results from Figure 1 and Figure 2 show that TEXTRUNNER's precision is low when the CRF confidence in the extraction is low, when the redundancy of the extraction is low, and when the arguments are far apart. Thus, system confidence, redundancy, and locality form the key factors for our filter policy: if the confidence is less than 0.5 and the redundancy is less than 2 or the distance between the arguments in the sentence is greater than 5 (if the relation is binary) or 8 (if the relation is n-ary) discard this tuple. These thresholds were determined by a parameter search over a small dataset.

Order of Sentences for Extraction: An optimal ordering policy would apply SRL-IE first to the sentences where TEXTRUNNER has low precision and leave the sentences that seem malformed (*e.g.*, incomplete sentences, two sentences spliced together) for last. As we have seen, the distance between the first and last argument is a good indicator for TEXTRUNNER precision. Moreover, a confidence value of 0.0 by TEXTRUNNER's CRF classifier is good evidence that the sentence may be malformed and is unlikely to contain a valid relation.

We rank sentences S in the following way, with SRL-IE processing sentences from highest ranking to lowest: *if* CRF.confidence = 0.0 *then* S.rank = 0, *else* S.rank = average distance between pairs of arguments for all tuples extracted by TEXTRUNNER from S.

While this ranking system orders sentences according to which sentence is likely to yield maximum new information, it misses the cost of computation. To account for computation time, we also estimate the amount of time SRL-IE will take to process each sentence using a linear model trained on the sentence length. We then choose the sentence that maximizes information gain divided by computation time.

6.1 Properties of Hybrid Extractors

The choice between the two hybrid systems is a trade-off between recall and precision: RECALLHYBRID guarantees the best recall, since it does not lose any extractions, while PRECHYBRID is designed to maximize the early boost in precision. The evaluation in the next section bears out these expectations.

6.2 Evaluation of Hybrid Extractors

Figure 3(a) and Figure 4(a) report the precision of each system for binary and n-ary extractions measured against available computation time. PRECHYBRID starts at slightly higher precision due to our filtering of potentially low quality extractions from TEXTRUNNER. For binary this precision is even better than SRL-IE's. It gradually loses precision until it reaches SRL-IE's level. RECALLHYBRID improves on TEXTRUNNER's precision, albeit at a much slower rate and remains worse than SRL-IE and PRECHYBRID throughout.

The recall for binary and n-ary extractions are shown in Figure 3(b) and Figure 4(b), again measured against available time. While PRECHYBRID significantly improves on TEXTRUNNER's recall, it does lose recall compared to RECALLHYBRID, especially for n-ary extractions. PRECHYBRID also shows a large initial drop in recall due to filtering.

Lastly, the gains in precision from PRECHYBRID are offset by loss in recall that leaves the F1 measure essentially identical to that of RECALLHYBRID (Figures 3(c),4(c)). However, for a fixed time budget both hybrid F-measures are significantly better than TEXTRUNNER and SRL-IE F-measures demonstrating the power of the hybrid extractors.

Both methods reach a much higher F1 than TEXTRUNNER: a gain of over 0.15 in half SRL-IE's processing time and over 0.3 after the full processing time. Both hybrids perform better than SRL-IE given equal processing time.

We believe that most often constructing a higher quality database of facts with a relatively lower recall is more useful than vice-versa, making PRECHYBRID to be of wider applicability than RECALLHYBRID. Still the choice of the actual hybrid extractor could change based on the task.

7 Related Work

Open information extraction is a relatively recent paradigm and hence, has been studied by only a small number of researchers. The most salient is TEXTRUNNER, which also introduced the model (Banko et al., 2007; Banko and Etzioni, 2008).

A version of KNEXT uses heuristic rules and syntactic parses to convert a sentence into an unscoped logical form (Van Durme and Schubert, 2008). This work is more suitable for extracting common sense knowledge as opposed to factual information.

Another Open IE system, Kylin (Weld et al., 2008), suggests automatically building an extractor for each relation using self-supervised training, with training data generated using Wikipedia infoboxes. This work has the limitation that it can only extract relations expressed in Wikipedia infoboxes.

A paradigm related to Open IE is Preemptive IE (Shinyama and Sekine, 2006). While one goal of Preemptive IE is to avoid relation-specificity, Preemptive IE does not emphasize Web scalability, which is essential to Open IE.

(Carlson et al., 2009) presents a semi-supervised approach to information extraction on the Web. It learns classifiers for different relations and couples the training of those classifiers with ontology defining constraints. While we attempt to learn unknown relations, it learns a pre-defined set of relations.

Another related system is WANDERLUST (Akbik and Broß, 2009). The authors of this system annotated 10,000 sentences parsed with LinkGrammar, resulting in 46 general linkpaths as patterns for relation extraction. With these patterns WANDERLUST extracts binary relations from link grammar linkages. In contrast to our approaches, this requires a large set of hand-labeled examples.

USP (Poon and Domingos, 2009) is based on Markov Logic Networks and attempts to create a full semantic parse in an unsupervised fashion. They evaluate their work on biomedical text, so its applicability to general Web text is not yet clear.

8 Discussion and Future Work

The Heavy Tail: It is well accepted that information on the Web is distributed according to Zipf's

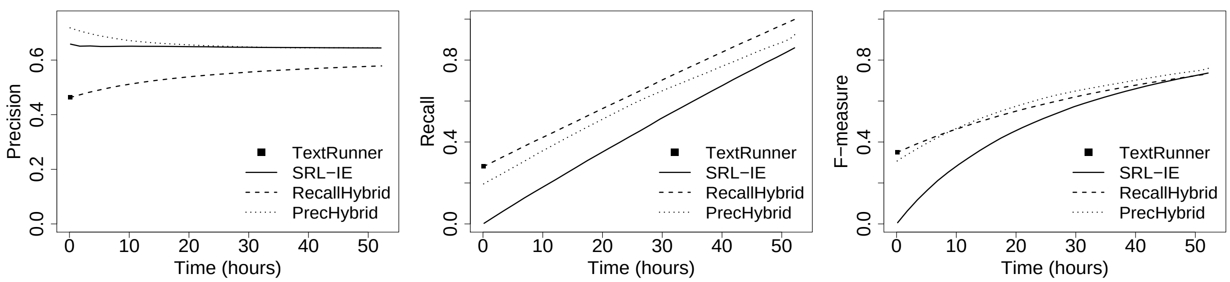

Figure 3: (a) Precision for binary extractions for PRECHYBRID starts higher than the precision of SRL-IE. (b) Recall for binary extractions rises over time for both hybrid systems, with PRECHYBRID starting lower. (c) Hybrid extractors obtain the best F-measure given a limited budget of computation time.

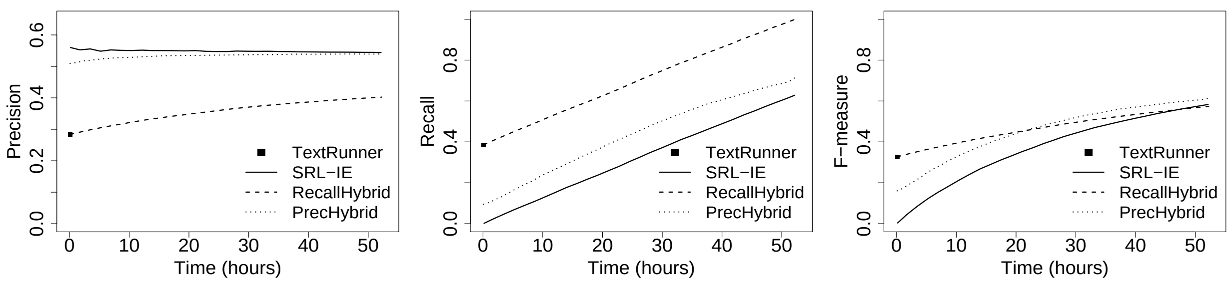

Figure 4: (a) PRECHYBRID also gives a strong boost to precision for n-ary extractions. (b) Recall for n-ary extractions for RECALLHYBRID starts substantially higher than PRECHYBRID and finally reaches much higher recall than SRL-IE alone. (c) F-measure for n-ary extractions. The hybrid extractors outperform others.

Law (Downey et al., 2005), implying that there is a heavy tail of facts that are mentioned only once or twice. The prior work on Open IE ascribes prime importance to redundancy based validation, which, as our results show (Figures 1(b), 2(b)), captures a very tiny fraction of the available information. We believe that deeper processing of text is essential to gather information from this heavy tail. Our SRL-IE extractor is a viable algorithm for this task.

Understanding SRL Components: UIUC-SRL as well as other SRL algorithms have different subcomponents – parsing, argument classification, joint inference, *etc*. We plan to study the effective contribution of each of these components. Our hope is to identify the most important subset, which yields a similar quality at a much reduced computational cost. Another alternative is to add the best performing component within TEXTRUNNER.

9 Conclusions

This paper investigates the use of semantic features, in particular, semantic role labeling for the task of open information extraction. We describe SRL-IE, the first SRL based Open IE system. We empirically compare the performance of SRL-IE with TEXTRUNNER, a state-of-the-art Open IE system and find that on average SRL-IE has much higher recall and precision, however, TEXTRUNNER outperforms in precision for the case of highly redundant or high locality extractions. Moreover, TEXTRUNNER is over 2 orders of magnitude faster.

These complimentary strengths help us design hybrid extractors that achieve better performance than either system given a limited budget of computation time. Overall, we provide evidence that, contrary to belief in the Open IE literature (Banko and Etzioni, 2008), semantic approaches have a lot to offer for the task of Open IE and the vision of machine reading.

10 Acknowledgements

This research was supported in part by NSF grant IIS-0803481, ONR grant N00014-08-1-0431, and DARPA contract FA8750-09-C-0179, and carried out at the University of Washington's Turing Center.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*.

Alan Akbik and Jügen Broß. 2009. Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *Proceedings of the Workshop on Semantic Search (SemSearch 2009) at the 18th International World Wide Web Conference (WWW 2009)*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics*, pages 86–90.

Michele Banko and Oren Etzioni. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2670–2676.

Michele Banko. 2009. *Open Information Extraction for the Web*. Ph.D. thesis, University of Washington.

Andrew Carlson, Justin Betteridge, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2009. Coupling semi-supervised learning of categories and relations. In *Proceedings of the NAACL HLT 2009 Workskop on Semi-supervised Learning for Natural Language Processing*.

Bonaventura Coppola, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Shallow semantic parsing for spoken language understanding. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 85–88.

Doug Downey, Oren Etzioni, and Stephen Soderland. 2005. A probabilistic model of redundancy in information extraction. In *IJCAI '05: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1034–1041.

Oren Etzioni, Michele Banko, and Michael J. Cafarella. 2006. Machine reading. In *AAAI'06: proceedings of the 21st national conference on Artificial intelligence*, pages 1517–1519.

Richard Johansson and Pierre Nugues. 2008. The effect of syntactic representation on semantic role labeling. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 393–400.

Paul Kingsbury Martha and Martha Palmer. 2002. From treebank to propbank. In *In Proceedings of LREC-2002*.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree kernels for semantic role labeling. *Computational Linguistics*, 34(2):193–224.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10.

V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311.

Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1-3):233–272.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191.

Benjamin Van Durme and Lenhart Schubert. 2008. Open knowledge extraction through compositional language processing. In *STEP '08: Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 239–254.

Daniel S. Weld, Raphael Hoffmann, and Fei Wu. 2008. Using wikipedia to bootstrap open information extraction. *SIGMOD Rec.*, 37(4):62–68.