Discourse-Wide Extraction of Assay Frames from the Biological Literature

Dayne Freitag
SRI International
9988 Hibert Street, Suite 203
San Diego, CA 92131, USA
freitag@ai.sri.com

Paul Kalmar
SRI International
9988 Hibert Street, Suite 203
San Diego, CA 92131, USA
paul.kalmar@sri.com

Eric Yeh
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025, USA
yeh@ai.sri.com

Abstract

We consider the problem of populating multi-part knowledge frames from textual information distributed over multiple sentences in a document. We present a corpus constructed by aligning papers from the cellular signaling literature to a collection of approximately 50,000 reference frames curated by hand as part of a decade-long project. We present and evaluate two approaches to the challenging problem of reconstructing these frames, which formalize biological assays described in the literature. One approach is based on classifying candidate records nominated by sentence-local entity co-occurrence. In the second approach, we introduce a novel virtual register machine that traverses an article and generates frames, trained on our reference data. Our evaluations provide evidence that best performance in the task ultimately hinges on an integration of information distributed over multiple sentences.

1 Introduction

Biological event and relation extraction have been the focus of considerable study in recent years, resulting in the availability of annotated corpora (Kim et al., 2003; Pyysalo et al., 2007; Kim et al., 2008; Thompson et al., 2009). In the interest of replicability and progress on critical challenges, such resources typically decompose the hard problem of factual understanding into several simpler problems, such as entity recognition, binary relation detection, and co-reference resolution.

This methodology is subject to several criticisms. The reliance on thorough annotation imposes overheads that prevent rapid progress. The targeting of a fixed set of simplified, typically binary relations does justice neither to the complexity of information expressed in a typical sentence, nor to the biological processes under discussion. And the methodology places an emphasis on pieces of information amenable to expression in individual sentences, leaving untouched information that can be assembled only through traversal of multiple sentences.

In this paper we address the problem of constructing multi-slot knowledge frames from the technical literature on cellular signaling networks. The frames in our study are a faithful representation of assays reported in this literature, called *datums*, with only approximate localization to specific textual regions. We have no one-to-one mapping between frames and sentences, no guarantee that the slots of a frame co-occur in a single sentence, and no universal presentational convention governing the sequence of slot-relevant expressions. Nevertheless, we seek to learn procedures for populating frames in new documents.

Success in this endeavor would have significant practical impact. If we can automate the separation of experimental evidence from common knowledge and speculation, we have the means to construct a high-quality biomedical resource of use to both experimental and computational biologists. Our efforts, for example, ultimately seek to automate the maintenance and extension of high-fidelity machine models of signaling pathways associated with Ras-driven human cancer.

We offer three contributions. First, we describe a problem of clear biomedical significance that involves synthesis of information distributed across a document, one that poses pertinent challenges to the current practice of machine reading. Second, we describe and evaluate an approach (the *frame classification* approach) that formalizes this problem as a binary classification of frames nominated

Proceedings of the Biomedical NLP Workshop associated with RANLP 2017, pages 15–23,
Varna, Bulgaria, 8 September 2017.
https://doi.org/10.26615/978-954-452-044-1_003

by protein pairs co-occurring in sentences. We provide evidence that good performance on this problem requires attention to how entities are referenced across a document, even in multiple documents, not just in the nominating sentence. Finally, we describe and evaluate an approach (the *register machine* approach) that attempts to correct deficiencies of the frame classification approach, specifically its limiting reliance on sentence-local juxtaposition of frame slot elements. This approach formalizes the frame extraction problem as learning the best sequence of instructions for frame generation through document traversal.

2 Related Work

Progress in biomedical information extraction (BioIE) is measured against shared annotated corpora that decompose the problem into entity extraction and sentence-level relation and event detection (Kim et al., 2003; Pyysalo et al., 2007; Kim et al., 2008; Thompson et al., 2009). The structure of these tasks has remained remarkably stable over the years, differing in some important ways from the task addressed in this paper. Most notably, the canonical BioIE task is highly localized and mostly agnostic to discourse context. The objective is to determine whether a single sentence expresses some event of interest–gene expression, phosphorylation, regulation, etc.–and, if so, what roles the entities appearing in the sentence play in the putative event. The "events" detected in this fashion are divorced from their discourse context (modulo coreference resolution), although some attention has been paid to epistemic qualifications, such as negations and speculations (Kim et al., 2011). We have posed ourselves a more focused task—extract the experiments described in a paper—and are forced to do without reliable sentence-level annotation.

There is no doubt that our system must respond to some of the same expressions that are addressed in some of these shared tasks. In particular, the Genia Event Extraction Task (Kim et al., 2011) targets phosphorylation and regulation events involving phosphorylation, among other things. Many of these event mentions are encountered in sections detailing experiments. Thus, our task can be addressed in part through disambiguation and assimilation of these events—which were actually observed in experiments? In this paper, we describe an approach to datum extraction that elaborates this idea.

Our focus on multi-slot, multi-sentence factual frames is reminiscent of early formulations of the information extraction problem used in the Message Understanding Conference (MUC) (Grishman and Sundheim, 1996; Chinchor et al., 1993). Over successive iterations of MUC, target frames became quite elaborate, similar in complexity to the datums we ultimately seek to populate. Many of the tasks that the information extraction community views as canonical, including named entity recognition, co-reference resolution, word sense disambiguation, and relation and event extraction, were introduced as simplifications of the core frame-filling task in MUC6. The field has since largely neglected the discourse-wide frame-filling challenge.

Of course, to take it up again and address it with the latest machine learning techniques, we require heuristics to align slot-level information found in reference frames to expressions in training sentences. Using such heuristics, in combination with structured ground-truth data, such as our collection of datums, is commonly referred to as *distant supervision*, an approach pioneered on a biomedical extraction problem (Craven and Kumlien, 1999). Relatively little work has applied distant supervision to discourse-level extraction problems. A counter-example is Reschke et al. (2014), which addresses event extraction at the document level, attempting to populate event-related Wikipedia "info boxes" with article source text. The Reschke et al. approach employs SEARN (Daume III et al., 2009), a technique that reduces complex structured classification problems into simpler sequence learning problems, finding that it yields performance superior to several strong baselines.

Recently, the problem of understanding accounts of experiments in the biological literature has been the focus of a small amount of study (Dasigi et al., 2017; Burns et al., 2016). This work, which springs from the same motivation and shares some of the same data as our own, is largely concerned with modeling the discourse structure of experimental narratives. It is therefore largely complementary to our work, which targets factual experimental details. Success in discourse modeling promises to solve key problems that we face, such as the segmentation of the text into distinct experiments.

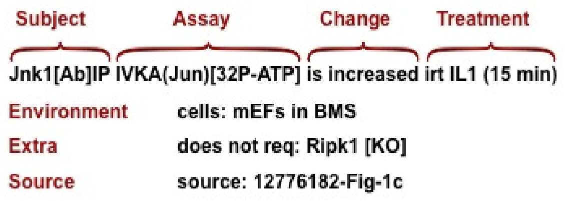

Figure 1: An annotated Pathway Logic datum.

3 Problem

The Pathway Logic (PL) project pursues high-fidelity signaling pathway models centering on the Ras family of proteins (Eker et al., 2004). Part of the effort involves a manual curation of experimental results, which has resulted in approximately 50K records, each containing a detailed formal representation of a reported experiment and its outcomes. Such records, called *datums*, retain pointers to the papers and figures from which they were derived.

Figure 1 displays a typical datum in its compact formal syntax, highlighting the four key components: the *assay*, encoding the type of assay conducted (here, an in-vitro kinase activity assay); the *subject*, the entity whose response was measured ("Jnk1"); the *treatment*, the substance applied to the cellular environment (here, some member of the IL-1 family, either IL-1 alpha or IL-1 beta); and the *change* or experimental outcome. It should be apparent from the figure that the typical datum records many additional experimental details. We refer to the combination of these four fields, stripped of such qualifiers, as a *simplified datum*, and seek to reconstruct these 4-tuples in our experiments.

Notable among the fields in Figure 1, is an encoding of the source of the datum, most frequently as a PubMed ID and figure reference. The datum curator, not a computational linguist, found it most natural to localize datums to the figures displaying assay outcomes. As a consequence, we do not have access to a simple procedure for identifying specific textual expressions for the various datum elements. In fact, the data comes with no guarantee that such expressions are present at all.

However, after a manual review of a large number of datums, we know that while some datums are not adequately described in the text of annotated articles, most are. Furthermore, the alignment of datums to figures enables weak localization of datum elements to individual sentences, because figure captions and body sentences containing figure references are on average relatively rich in information needed to populate the simplified datums attached to corresponding figures.

1. *Phosphorylation and activation of JAK1 and Stat6 are essential for induction of Stat6 DNA binding activity.* 2. *To ascertain whether the decrease in Stat6 DNA binding activity in the SOCS-1 stable transfectants was due to inhibition of JAK1 kinase activity, we immunoprecipitated lysates from cells untreated or treated with IL-4 with Abs to JAK1 or Stat6 and probed with Ab to phospho-tyrosine.* 3. *Induction of JAK1 and Stat6 phosphorylation in the SOCS-1 stable clones was reduced when compared with control (Fig. 3A), while induction in the SOCS-2 stable clones (Fig. 3B) and in the SOCS-3 stable clones (Fig. 3C) was similar to that of controls.* 4. *To further confirm that SOCS-1 suppresses JAK1 activation, we measured the IL-4-induced kinase activity of JAK1 in the SOCS-1 stable clones by in vitro kinase assay.*

Table 1: Example sentences potentially expressing the key elements of a "phos" datum.

Table 1, which excerpts four contiguous sentences (we have numbered them for convenience) from a relevant article, renders this concrete, but also illustrates some of the subtleties involved. Our data notes two distinct phosphorylation assays in this passage, both linked to Sentence 3 (the only sentence with figure references), corresponding to the subjects JAK1 and Stat6, respectively, each of which are phosphorylated (i.e., the change of the "phos" assay is "increased") in response to IL-4 (the treatment).

This passage is abundant in evidence about the relevant experiments, but the information is distributed. Sentences 1 and 3 both contain inflections of "phosphorylate," providing evidence that a "phos" assay was conducted, but both lack the treatment IL-4, which is referenced in Sentences 2 and 4. Note that the "phosphorylate" sentences are rich in entities, potentially posing a combinatoric discrimination problem. Ultimately, if we wish to extract the two target datums at Sentence 3, information about the experimental treatment must be pulled in from one of the adjacent sentences, and we must determine that exactly two datums are warranted.[1]

[1] Actually, a number of experimental variants are under discussion in this passage. These are captured the database in supplementary records called "extras." Extras are not the

Whatever the textual evidence for datums in a paper, our problem is essentially extraction at the level of documents. Formally, we are given a set of examples $\{\langle d_i, y_i \rangle\}$, in which d_i is a document and y_i is a set of tuples $\{\langle s_{ij}, t_{ij}, a_{ij}, c_{ij} \rangle\}$, the elements of each tuple representing subjects, treatments, assay types, and observed changes, respectively. Assay and change values are drawn from closed classes, assays from the set of types represented in the Pathway Logic knowledge base, and changes from the set {*increased*, *decreased*, *unchanged*}. Subjects and treatments are drawn from the effectively open class of chemicals used for experiments in the literature. In practice, they are usually proteins, and in our experiments these two slots take Uniprot IDs. Our extraction task involves inferring the correct set y_i, given some d_i.

4 Approaches

We investigate two distinct approaches to this problem, the *frame classification* approach and the *register machine* approach. The first is applicable only to subject-treatment pairs that co-occur in individual sentences, while the second approach can in principle associate subjects and treatments found in different sentences.

4.1 Frame Classification

We observe that datum subjects and treatments tend to be mentioned together in individual sentences. This motivates a simple framing of the datum extraction problem as binary classification. Specifically, if we fix the assay type (e.g., "phos") and change (e.g., "increased"), we can view each document as a set of co-mentioned proteins—all pairs of proteins mentioned together in some sentence—and attempt to distinguish pairs in the subject-treatment relation from other pairs. Of course, we must perform this procedure for all assay-change pairs of interest.

We follow an approach to featurization proposed in Xu et al (2016). Consider the set of sentences containing protein entities P_1 and P_2. Given a target assay-change configuration we train *two* binary classification models, one to distinguish cases in which P_1 and P_2 are subject and treatment, respectively, and one for the opposite assignment. Our feature vectors have four parts, each part containing features that require the frequency of lexical unigrams and bigrams found in various sentence contexts. Thus, the word "protein" corresponds to three distinct features: one feature recording its frequency of occurrence before P_1 in the set of sentences, between P_1 and P_2 (encountered in that order), between P_2 and P_1 (encountered in that order), and after P_2, respectively.

We also included and recorded a small performance benefit from two non-lexical features. First, observing that datum protein pairs tend to be more frequent than others, we defined a feature that reflects the number of sentences in which a pair co-occurs. Second, we defined indicator features that reflect whether specific proteins fill a subject or treatment role anywhere in the training data.

Admittedly, this approach suffers from certain limitations, most obviously limited recall, as it can only distinguish datums whose subject and treatment co-occur in a sentence—e.g., discarding some 40% of phosphorylation datums. And as noted, because the classification problem is conditioned on assay and change, we must learn a separate classifier for each observed assay-change combination. This is tractable in practice, because the number of frequently observed assay-change combinations occurring is manageable.

4.2 Register Machine

To accommodate the distribution of relevant information across the sentences in a discourse, we imagine a model capable of traversing sentences, accumulating information, and synthesizing datums. We suppose that datums are produced by a virtual machine with four registers (one for each of the slots in a simplified datum) and two cursors (to traverse the sentences in a caption and article body, respectively). At each time step, the machine can execute an instruction to advance either cursor, populate or delete the contents of registers, or produce one or more datums. Specifically, we define the following instructions:

- **advance*Section*Cursor**, where *Section* can be either *Caption* or *Body*. One of the cursors is advanced to the next entity within the current sentence, if present, or to the beginning of the next sentence in body or among figure captions.

- **set*ClosedValue***, where *Closed* can be either *Assay* or *Change*, and *Value* is one of the legal values for the indicated closed-class register. The register becomes populated with

focus of this paper's work, but are ultimately important.

the specified value, replacing any previous contents.

- **set*Open*from*Section***, where *Open* is either *Subject* or *Treatment*. The indicated register is populated with the entity under the cursor for *Section*. This instruction is illegal if there is no such entity.

- **add*Open*from*Section***. This instruction is like the previous one, except the entity is accumulated into the indicated register. As this implies, open-class registers can hold multiple entities.

- **delete*Register*** empties the indicated register.

- **deleteAll** empties all registers.

- **produceDatums** causes datums to be generated from register contents. A different datum is generated for each distinct combination of entities in the subject and treatment registers.

Let us suppose we are given a sequence of instructions $I = i_1 \cdots i_m$ applying the machine to some example $\langle d_i, y_i \rangle$. It is easy to see than any such I yields a set of datums y_i^*, which we can formalize as some function, $F(d, I) = y$.[2] Further, we can speak of a policy $\pi(d) = I$ that nominates instructions sequences, given a document. Ultimately, our objective is to find the best policy:

$$\underset{\pi}{\operatorname{argmin}} \sum_i L(F(d_i, \pi(d_i)), y_i) \qquad (1)$$

Here, $L(y^*, y)$ is the loss experienced by some machine-generated set of datums y^* with respect to the ground-truth y. In practice, we seek to optimize the F1 of extracted datums versus ground truth under a strict equality standard, i.e., only those datums that agree in all slots with some ground-truth datum are counted as successes.

Of course, Equation 1 is difficult to satisfy directly. Instead, we seek to learn a local ranking model for individual instructions. Let $S(d, I_{1,k})$ represent the state of the machine after executing k instructions $I_{1,k} = i_1 \cdots i_k$ against document d, including the positions of the cursors, the state of the registers, and any generated datums. We seek to learn a local policy $\hat{\pi}(S(d, I_{1,k})) = i_{k+1}$ that chooses the best next instruction.

Learning $\hat{\pi}$ is essentially a ranking problem: given all legal instructions in the current state, which is best to execute? We therefore adopt a learning-to-rank approach, training an empirical model to map machine states to real values, such that the highest-scoring instruction is the best to execute in the current state. To this end, we implemented an oracular policy (henceforth the "oracle") that nominates instructions based on full knowledge of ground truth. Given our uncertainty about which sentences express datum elements (the subject of one or more datums might be mentioned dozens of time in an article), this policy heuristically orients datum production around figure captions and sentences containing figure references: datums are aligned to such sentences, using their source field, and the machine is instructed to load its registers and produce datums as close as possible to the sentences identified in this way. For example, if a datum having subject a and treatment b is linked to sentence s_i, and b is mentioned in s_i, but the nearest mention of a is in s_{i-1}, the oracle instructs the machine to load its subject register at the a mention in s_{i-1}, and its treatment, assay, and change registers at the b mention in s_i, followed by a `produceDatums` instruction (and typically some combination of `delete` instructions).

In our current implementation, we train a multiclass perceptron model to perform ranking, updating it whenever it ranks an inappropriate instruction highest. The mistake-driven nature of this training regime enables us to accommodate a subtlety of the problem: there are often several good instructions in any given state, and we cannot know that the instruction preferred by the oracle is truly optimal. To respond to this reality, the oracle provides a second service—assessment of instructions preferred by the model. If such an instruction is deemed adequate—if it does not ultimately prevent the register machine from producing upcoming datums—the model's preferred instruction is deemed correct, and no update is performed.

Any feature of the machine's state, including the contents of its registers, datums produced so far, recently executed instructions, and, most importantly, the language at and around cursors, may be encoded to train the model. Table 2 lists the features implemented to date, which should be self-explanatory, except for the "Pattern" features. To implement these, we separately induce a set of patterns over dependency parses to detect expressions

[2] We posit that illegal instructions (e.g., advancing a cursor at the end of the document) have no effect.

Type	Feature	Description
Cursor	`atPosition(curs, pos)`	True if the cursor `curs` (body or caption) is at the indicated `pos` in its section (beginning, internal, end)
Register	`populated(reg)`	True if the indicated `reg` (subject, treatment, assay, or change) is populated.
	`cregContains(reg, val)`	True if a closed-class register `reg` (assay or change) contains a particular value `val` legal for that type (e.g. the assay register contains "phos").
	`oregContains(curs, reg)`	True if the open-class register `reg` contains the entity under the cursor `curs`.
	`allPopulated`	True if all four registers are populated.
Lexical	`sentContains(curs, word)`	True if the sentence under `curs` contains `word`.
	`wordAtOffset(curs, offs, word)`	True if `word` is observed at offset `offs`, ranging over $[-2, +2]$, from `curs`.
Pattern	`activeAtSent(pat, curs)`	True if the detection pattern `pat` matches the sentence under `curs`.
	`activeAtEnt(pat, curs)`	True if the pattern `pat` matches the entity under `curs`.
Other	`producedDatums`	True immediately after a produceDatums instruction has been executed.
	`bias`	Always true.

Table 2: Features used in experiments with the register machine.

that tend to signal the presence of an assay subject or treatment (Freitag and Niekrasz, 2016). For each such pattern, we define two features, which are true if the corresponding pattern matches anywhere in a cursor sentence or at a cursor entity, respectively. In addition to the features listed in the table, we automatically generate a large number of conjunctive features from feature pairs, returning true when both the constituent features are true.

	All	Phos
Database	17,444	4,864
Experimental corpus	6,554	3,152
Visible	5,981	2,989
Fully visible	2,336	1,418

Table 3: Visibility of datums (of any type vs. those representing phosphorylation assays).

5 Evaluation

We constructed our experimental data from the set of datums in the Pathway Logic database, along with the 2,394 papers to which they refer. Because most of these papers are available only as PDF,[3] we converted them to plain text and heuristically identified paper sections, converting each to a sequence of sentences. This data was then annotated by machine to identify mentions of protein entities (heuristically mapped to Uniprot identifies) and figure references. The latter were used to align datums heuristically to sentences.

As noted previously, the Pathway Logic data comes with no guarantee that the datums are actually described in the text of an article.[4] Moreover, failures in entity recognition or resolution further reduce what our models have the potential to "see" in the text. We therefore limit our attention to "visible" datums, those datums for which we recognize either the subject or treatment entity somewhere in the paper to which a datum is aligned. We call datums for which both entities are recognized "fully visible." Our experimental corpus consists of the 518 papers aligned to at least one visible datum.

[3] Much of the curated data predates the establishment of the NXML format.

[4] Nor is there a strong guarantee that all experiments described in a paper have been converted into datums. Our curator has it in her charter to do so, but we have encountered experiments for which no datum was created. We do not know how common this is.

Method	Precision	Recall	F1
Oracle	0.6935	0.5708	0.5996
Frame Gold	0.9302	0.4937	0.6017
Frame	0.2426	0.296	0.2322
Machine	0.2056	0.1877	0.165

Table 4: Macro-averaged precision, recall, and F1 in extracting simplified "phos" datums.

Table 3 provides an overview of the data we work with. For convenience in comparing our two approaches, we focus on "phos" datums exclusively, and therefore present separate totals for "phos" datums in the table. (The register machine targets all visible assay types, but we evaluate its performance only against "phos" datums.) The row labeled *Database* lists counts calculated from our snapshot of the datum database, while *Experimental corpus* considers only the subset aligned to papers in our collection of 518 articles. The rows *Visible* and *Fully visible* document the number of datums actually available for experiments. The performance numbers that follow correspond to those datums contained in the cell labeled *Visible, Phos*.

In our experiments, we randomly sampled 75% of our 518 articles (and the corresponding datums) for training, and evaluated against the remaining 25%. In training the register machine, we reserve some of the training data for validation, using F1 against this hold-out data as a stopping criterion to prevent overfitting. To be deemed correct, an extracted simplified datum must agree with a ground-truth datum on all four slots. When a ground-truth datum is partially visible, an extractor must populate the empty slot with a null in order to be awarded credit. Note that this necessarily limits the recall of frame classification, which has no way to produce a null slot.

Table 4 presents the results of our experiments. The first two rows in the table establish approximate upper bounds on performance. *Oracle* measures the performance of the policy used to generate training data for the register machine, while *Frame Gold* lists the performance of a perfect classifier of candidate protein pairs nominated using the sentence co-occurrence heuristic. Interestingly, the difference in recall between these two approaches is fairly small, indicating that although the register machine can in principle integrate evidence distributed over multiple sentences, it is difficult to do so, even for a heuristically implemented oracle.

The remaining two rows compare the two learning approaches to datum extraction described in the paper, frame classification (*Frame*) and the register machine (*Machine*). Note that the example generation procedure used in Frame leads to considerable class skew, with the set of negative example dwarfing the positive. In these experiments, we randomly sampled the negative examples to achieve a ten-to-one negative-to-positive ratio.

The results appear to suggest that the relative simplicity of frame classification more than compensates for the fact that it cannot account for a significant fraction of datums, those whose subjects and treatments are not found together in an individual sentence. We see clear evidence that accumulation of evidence spread across sentences enhances performance. In a separate experiment, in which we classified individual sentences (similar to canonical relation extraction), we saw a drop in F1 of about 2 points.

The register machine, which in principle can accommodate the "distributed" datums that the frame classifier ignores, has difficulty learning instruction sequences well enough to achieve comparable performance. Its difficulty appears to center primarily on the extracted components of datums, the subjects and treatments. If we evaluate the register machine's performance on individual slots (e.g., by scoring the set of phos subjects extracted against the set found in phos datums aligned to a paper), we observe F1s of 0.92 and 0.67 on assay and change, respectively, but only 0.42 and 0.38 on subject and treatment. We believe that the feature set currently employed by the machine is too impoverished to perform these extractions accurately. Note that while frame classification accumulates evidence relevant to a protein pair from across an article, the register machine relies on mostly local information. This is an unnecessary limitation, which we are attempting to rectify.

6 Discussion

Our work with the frame classifier is leading the way in this regard. In preliminary work conducted after the experiments presented here, we have continued to mitigate keys drawbacks of the approach. For example, by training individual protein classifiers for “subjectness” and “treatmentness,” using information distributed across an article, we observe a frame classification F1 of 0.30 in preliminary experiments. We are also working to increase the number of assay-change combinations targeted by frame classification to practical levels.

All this makes clear that the strict evaluation metric used in this paper–simultaneous agreement on four key slots with target datums–poses a stiff challenge for computer readers. These performance levels are understandable. Robust solutions for many types of *binary* relation and event extraction have yet to be reported. For example, a characteristic approach to ACE-style relation extraction reports peak F1 of about 0.55 (GuoDong et al., 2005), and recent work in comparable biomedical extraction problems yields qualitatively comparable performance–e.g., F1 of 0.53 in a pathway curation task involving primarily binary interactions having high domain overlap with the current paper (Nédellec et al., 2013). As a rule, adding slots to a target template leads to considerably lowered extraction performance under a strict matching regime. Moreover, the heuristic alignment of slot values to specific textual expressions adds further noise to the training and evaluation processes.

However, there is reason to believe that even these modest performance numbers are useful for certain applications. In separate work under the DARPA Big Mechanism program, we implemented a manual datum extractor, as part of a system that sought to confirm events and relations extracted by general-purpose bio-NLP readers by looking for corroborative experiments in the same paper. We were able to show, using hand-scored results from the program evaluation, that 80% of machine extractions corroborated in this way were correct (about 17% of all such extractions), versus a baseline accuracy of 50%. This despite the fact that we estimated the F1 of our the hand-authored system, which over-generates wildly, at less than 0.02. Thus, even a very noisy experiment extractor has value as a source of corroboration for assertions extracted without attention to pragmatic context. Possibly key to this outcome was the strict standard applied in the program evaluation, which deprecated speculation or statements of background knowledge.

Our focus on a very specific problem and data set may leave the impression that these results are of little further use. We argue that the opposite is true, that this admittedly domain-specific challenge is an instance of a type of problem that will become increasingly salient as machine reading matures. Eventually, the field must move beyond sentence-local, contextless, low-arity extraction to the full population of knowledge frames summarizing information relevant to important use cases. A key resource to this end will be “found” structured resources loosely attached to textual source material, such as the auxiliary data associated with biological publications with increasing frequency, or Wikipedia info-boxes summarizing events in newswire (Reschke et al., 2014). The field requires methods that exploit such resources for the interpretation of key facts in text.

7 Conclusion

The problem introduced in this paper—that of extracting faithful representations of experiments described in the biological literature—has two features that distinguish it from much of the work on biomedical NLP: (1) It is closely aligned to the needs of computational biology, stemming from research independent from and uninformed by NLP. And (2) it cannot be adequately addressed by models that target the information found in individual sentences in isolation. These two features make for a problem of considerable depth and importance, both for biology and NLP. While it is clear that we have not solved this problem with the approaches documented here, we have sketched two potential solutions and illuminated some of the specific challenges that remain.

Acknowledgments

This project was supported by the U.S. Army Research Office. The content of this paper does not necessarily reflect the position or policy of the U.S. Government. No official endorsement should be inferred.

References

Gully APC Burns, Pradeep Dasigi, Anita de Waard, and Eduard H Hovy. 2016. Automated detection of discourse segment and experimental types from the text of cancer pathway results sections. *Database* 2016.

Nancy Chinchor, David D Lewis, and Lynette Hirschman. 1993. Evaluating message understanding systems: an analysis of the third message understanding conference (muc-3). *Computational linguistics* 19(3):409–449.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*. volume 1999, pages 77–86.

P. Dasigi, G. A. P. C. Burns, E. Hovy, and A. de Waard. 2017. Experiment Segmentation in Scientific Discourse as Clause-level Structured Prediction using Recurrent Neural Networks. *ArXiv e-prints* .

Hal Daume III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine learning* 75(3):297–325.

Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, and Carolyn Talcott. 2004. Pathway logic: Executable models of biological networks. *Electronic Notes in Theoretical Computer Science* 71:144–161.

Dayne Freitag and John Niekrasz. 2016. Feature derivation for exploitation of distant annotation via pattern induction against dependency parses. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*. Association for Computational Linguistics, Berlin, Germany, pages 36–45.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 466–471.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 427–434.

J.-D. Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. GENIA corpusa semantically annotated corpus for bio-textmining. *Bioinformatics* 19(suppl 1):i180–i182.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics* 9(1):1.

Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. Overview of genia event task in bionlp shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*. Association for Computational Linguistics, Stroudsburg, PA, USA, BioNLP Shared Task '11, pages 7–15.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*. Association for Computational Linguistics Sofia, Bulgaria, pages 1–7.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjrne, Jorma Boberg, Jouni Jrvinen, and Tapio Salakoski. 2007. BioInfer: a corpus for information extraction in the biomedical domain. *BMC bioinformatics* 8(1):50.

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher D. Manning, and Daniel Jurafsky. 2014. Event Extraction Using Distant Supervision. In *LREC*. pages 4527–4531.

Paul Thompson, Syed A. Iqbal, John McNaught, and Sophia Ananiadou. 2009. Construction of an annotated corpus to support biomedical information extraction. *BMC bioinformatics* 10(1):1.

Jun Xu, Yonghui Wu, Yaoyun Zhang, Jingqi Wang, Hee-Jin Lee, and Hua Xu. 2016. Cd-rest: a system for extracting chemical-induced disease relation in literature. *Database* 2016.