NLPContributions: An Annotation Scheme for Machine Reading of Scholarly Contributions in Natural Language Processing Literature

Jennifer D'Souza
TIB Leibniz Information Centre for Science and Technology
Hannover, Germany
jennifer.dsouza@tib.eu

Sören Auer
TIB Leibniz Information Centre for Science and Technology & L3S Research Center
Hannover, Germany
soeren.auer@tib.eu

ABSTRACT

We describe an annotation initiative to capture the scholarly contributions in natural language processing (NLP) articles, particularly, for the articles that discuss machine learning (ML) approaches for various information extraction tasks. We develop the annotation task based on a pilot annotation exercise on 50 NLP-ML scholarly articles presenting contributions to five information extraction tasks 1. machine translation, 2. named entity recognition, 3. question answering, 4. relation classification, and 5. text classification. In this article, we describe the outcomes of this pilot annotation phase. Through the exercise we have obtained an annotation methodology; and found ten core information units that reflect the contribution of the NLP-ML scholarly investigations. The resulting annotation scheme we developed based on these information units is called NLPContributions.

The overarching goal of our endeavor is four-fold: 1) to find a systematic set of patterns of subject-predicate-object statements for the semantic structuring of scholarly contributions that are more or less generically applicable for NLP-ML research articles; 2) to apply the discovered patterns in the creation of a larger annotated dataset for training machine readers [18] of research contributions; 3) to ingest the dataset into the Open Research Knowledge Graph (ORKG) infrastructure as a showcase for creating user-friendly state-of-the-art overviews; 4) to integrate the machine readers into the ORKG to assist users in the manual curation of their respective article contributions. We envision that the NLPContributions methodology engenders a wider discussion on the topic toward its further refinement and development. Our pilot annotated dataset of 50 NLP-ML scholarly articles according to the NLPContributions scheme is openly available to the research community at https://doi.org/10.25835/0019761.

CCS CONCEPTS

• **General and reference** → *Computing standards, RFCs and guidelines*; • **Information systems** → **Document structure**; **Ontologies**; **Data encoding and canonicalization**.

KEYWORDS

dataset, annotation guidelines, semantic publishing, digital libraries, scholarly knowledge graphs, open science graphs

1 INTRODUCTION

As the rate of research publications increases [51], there is a growing need within digital libraries to equip researchers with alternative knowledge representations, other than the traditional document-based format, for keeping pace with the rapid research progress [3]. In this regard, several efforts exist or are currently underway for semantifying scholarly articles for their improved machine interpretability and ease in comprehension [19, 24, 38, 49]. These models equip experts with a tool for semantifying their scholarly publications ranging from strictly-ontologized methodologies [19, 49] to less-strict, flexible description schemes [24, 37], wherein the latter aim toward the bottom-up, data-driven discovery of an ontology. Consequently, knowledge graphs [1, 4] are being advocated as a promising alternative to the document-based format for representing scholarly knowledge for the enhanced content ingestion enabled via their fine-grained machine interpretability.

The automated semantic extraction from scholarly publications using text mining has seen early initiatives based on sentences as the basic unit of analysis. To this end, ontologies and vocabularies were created [14, 39, 46, 47], corpora were annotated [20, 32], and machine learning methods were applied [31]. Recently, scientific IE has targeted search technology, thus newer corpora have been annotated at the phrasal unit of information with three or six types of scientific concepts in up to ten disciplines [5, 16, 22, 33] facilitating machine learning system development [2, 8, 10, 34]. In general, a phrase-focused annotation scheme more directly influences the building of a scholarly knowledge graph, since phrases constitute knowledge graph statements. Nonetheless, sentence-level annotations are just as poignant offering knowledge graph modelers

better context from which the phrases are obtained for improved knowledge graph curation.

Over which, many recent data collection and annotation efforts [26–28, 36] are steering new directions in text mining research on scholarly publications. These initiatives are focused on the shallow semantic structuring of the instructional content in lab protocols or descriptions of chemical synthesis reactions. This has entailed generating annotated datasets via structuring recipes to facilitate their automatic content mining for machine-actionable information which are presented otherwise in adhoc ways within scholarly documentation. Such datasets inadvertently facilitate the development of machine readers. In the past, such similar text mining research was conducted as the unsupervised mining of Schemas (also called *scripts*, *templates*, or *frames*)—as a generalization of recurring event knowledge (involving a sequence of three to ten events) with various participants [40]—primarily over newswire articles [7, 11–13, 42–44]. They were a potent task at generalizing over similar but distinct narratives—can be seen as knowledge units—with the goal of revealing their underlying common elements. However, little insight was garnered on their practical task relevance. This has changed with the recent surface semantic structuring initiatives over instructional content. It has led to the realization of a seemingly new practicable direction that taps into the structuring of text and the structured information aggregation under Scripts-based knowledge themes.

Since scientific literature is growing at a rapid rate and researchers today are faced with this publications deluge [25], it is increasingly tedious, if not practically impossible to keep up with the progress even within one's own narrow discipline. The Open Research Knowledge Graph (ORKG) [4] is posited as a solution to the problem of keeping track of research progress minus the cognitive overload that reading dozens of full papers impose. It aims to build a comprehensive knowledge graph that publishes the research contributions of scholarly publications per paper, where the contributions are interconnected via the graph even across papers. At https://www.orkg.org/ one can view the contribution knowledge graph of a single paper as a summary over its key contribution properties and values; or compare the contribution knowledge graphs over common properties across several papers in a tabulated survey. Practical examples of the latter can be found accessible online at https://www.orkg.org/orkg/featured-comparisons. This practically addresses the knowledge ingestion problem for researchers. How? With the ORKG comparisons feature, researchers are no longer faced with the daunting cognitive ingestion obstacle from manually scouring through dozens of papers of unstructured content in their field. Where this process traditionally would take several days or months, using the ORKG contributions comparison tabulated view, the task is reduced to just a few minutes. Assuming the individual paper contributions are structured in the ORKG, they can then simply deconstruct the graph, tap into the aspects they are interested in, and can enhance it for their purposes. Further, they can select multiple such paper graphs and with the click of a button generate their tabulated comparison. For additional details on systems and methods beyond just the contribution highlights, they can still choose to read the original articles, but this time around equipped with a better selective understanding of which articles they should read in depth. Of-course scholarly article abstracts are intended

for this purpose, but they are not machine interpretable, in other words, they cannot be comparatively organized. Further, the unstructured abstracts representation still treats research as data silos, thus with this model, research endeavors, in general, continue to be susceptible to redundancy [23], lacking a meaningful way of connecting structured and unstructured information.

1.1 Our Contribution

In this paper, we propose a surface semantically structured dataset of 50 scholarly articles for their research contributions in the field of natural language processing focused on machine learning applications (the NLP-ML domain) across five different information extraction tasks to be integrable within the ORKG. To this end, we (1) identify sentences in scholarly articles that reflect research contributions; (2) create structured (subject,predicate,object) annotations from these sentences by identifying mentions of the contribution candidate term phrases and their relations; and (3) group collections of such triples, that arise from either consecutive or non-consecutive sentences, under one of ten core information units that capture an aspect of the contribution of NLP-ML scholarly articles. These core information units are conceptually posited as thematic scripts [40]. The resulting model formalized from the pilot annotation exercise we call the NLPContributions scheme.

It has the following characteristics: (1) via a contribution-centered model, it makes realistic the otherwise forbidding task of semantically structuring full-text scholarly articles—our task only needs a surface structuring of the highlights of the approach which often can be found in the Title, the Abstract, one or two paragraphs in the Introduction, and in the Results section; (2) it offers guidance for a structuring methodology, albeit still encompassing subjective decisions to a certain degree, but overall presenting a uniform model for identifying and structuring contributions—note that without a model, such structuring decisions may not end up being comparable across users and their modeled papers (see Figure 6); (3) the dataset is annotated in JSON format since it preserves relation hierarchies; (4) the annotated data we produce can be practically leveraged within frameworks such as the ORKG that support structured scholarly content-based knowledge ingestion. With the integration of our semantically structured scholarly contributions data in the ORKG, we aim to address the tedious and time-consuming scholarly knowledge ingestion problem via its contributions comparison feature. And further, by using the graph-based model, we also address the problem of scholarly information produced as data silos, as the ORKG connects the structured information across papers.

2 BACKGROUND AND RELATED WORK

Sentence-based Annotations of Scholarly Publications. Early initiatives in semantically structuring scholarly publications focused on sentences as the basic unit of analysis. In these sentence-based annotation schemes, all annotation methodologies [20, 32, 47, 48] have had very specific aims for scientific knowledge capture. Seminal works in this direction consider the CoreSC (Core Scientific Concepts) sentence-based annotation scheme [32]. This scheme aimed to model in finer granularity, i.e. at the sentence-level, concepts that are necessary for the description of a scientific investigation, while traditional approaches employ section names

serving as coarse-grained paragraph-level annotations. Such se-mantified scientific knowledge capture was apt at highlighting selected sentences within computer-based readers. In this application context, mere sectional information organization for papers was considered as missing the finer rhetorical semantic classifi-cations. E.g., in a Results section, the author may also provide some sentences of *background* information, which in a sentence-wise semantic labeling are called Background and not Results. As another sentence-based scheme is the Argument Zoning (AZ) scheme [48]. This scheme aimed at modeling the rhetorics around knowledge claims between the current work and cited work. They used semantic classes as "Own_Method," "Own_Result," "Other," "Previous_Own," "Aim," etc., each elaborating on the rhetorical path to various knowledge claims. This latter scheme was apt for citation summaries, sentiment analysis and the extraction of information pertaining to knowledge claims. In general, such complementary aims for the sentence-based semantification of scholarly publica-tions can be fused to generate more comprehensive summaries.

Phrase-based Annotations of Scholarly Publications. The trend towards scientific terminology mining methods in NLP steered the release of phrase-based annotated datasets in various domains. An early dataset in this line of work was the ACL RD-TEC cor-pus [22] which identified seven conceptual classes for terms in the full-text of scholarly publications in Computational Linguistics, viz. *Technology and Method*; *Tool and Library*; *Language Resource*; *Language Resource Product*; *Models*; *Measures and Measurements*; and *Other*. Similar to terminology mining is the task of scientific keyphrase extraction. Extracting keyphrases is an important task in publishing platforms as they help recommend articles to read-ers, highlight missing citations to authors, identify potential re-viewers for submissions, and analyse research trends over time. Scientific keyphrases, in particular, of type *Processes, Tasks* and *Materials* were the focus of the SemEval17 corpus annotations [5]. The dataset comprised annotations of the full text articles in Com-puter Science, Material Sciences, and Physics. Following suit was the SciERC corpus [33] of annotated abstracts from the Artificial Intelligence domain. It included annotations for six concepts, viz. *Task*, *Method*, *Metric*, *Material*, *Other-Scientific Term*, and *Generic*. Finally, in the realm of corpora having phrase-based annotations, was the recently introduced STEM-ECR corpus [16] notable for its multidisciplinarity including the Science, Technology, Engineering, and Medicine domains. It was annotated with four generic concept types, viz. *Process*, *Method*, *Material*, and *Data* that mapped across all domains, and further with terms grounded in the real-world via Wikipedia/Wiktionary links.

Next, we discuss related works that semantically model instruc-tional scientific content. In these works, the overarching scientific knowledge capture theme is the end-to-end semantification of an experimental process.

Shallow Semantic Structural Annotations of Instructional Content in Scholarly Publications. Increasingly, text mining ini-tiatives are seeking out recipes or formulaic semantic patterns to automatically mine machine-actionable information from scholarly articles [26–28, 36].

In [27], they annotate wet lab protocols, covering a large spec-trum of experimental biology, including neurology, epigenetics, metabolomics, cancer and stem cell biology, with actions corre-sponding to lab procedures and their attributes including materials, instruments and devices used to perform specific actions. Thereby the protocols then constituted a prespecified machine-readable for-mat as opposed to the ad-hoc documentation norm. Kulkarni et al. [27] even release a large human-annotated corpus of semantified wet lab protocols to facilitate machine learning of such shallow se-mantic parsing over natural language instructions. Within scholarly articles, such instructions are typically published in the Materials and Method section in Biology and Chemistry fields.

Along similar lines, inorganic materials synthesis reactions and procedures continue to reside as natural language descriptions in the text of journal articles. There is a growing impetus in such fields to find ways to systematically reduce the time and effort required to synthesize novel materials that presently remains one of the grand challenges in the field. In [26, 36], to facilitate machine learning models for automatic extraction of materials syntheses from text, they present datasets of synthesis procedures annotated with semantic structure by domain experts in Materials Science. The types of information captured include synthesis operations (i.e. predicates), and the materials, conditions, apparatus and other entities participating in each synthesis step.

The NLPContributions annotation methodology proposed in this paper draws on each of the earlier categorizations of related work. **First**, the full-text of scholarly articles including the Title and the Abstract are annotated in a sentence-wise granularity with the aim of the annotated sentences being only those restricted to the contributions of the investigation. We selectively consider the full-text of the article by focusing only on specific sections of the article such as the Abstract, Introduction, and the Results sections. Sometimes we also model the contribution highlights from the Approach/System description in case if the Introduction does not contain such pertinent information of the proposed model. We skip the Background, Related Work, and Conclusion sections altogether. These sentences are then grouped under one of ten main informa-tion units, viz. ResearchProblem, Objective, Approach, Tasks, ExperimentalSetup, Hyperparameters, Baselines, Results, and AblationAnalysis. Each of these units are defined in detail in the next section. **Second**, from the grouped contribution-centered sentences, we perform phrase-based annotations for (subject, predi-cate, object) triples to model in a knowledge graph. And **Third**, the resulting dataset has an overarching knowledge capture objective: capturing the contribution of the scholarly article and, in particular, to facilitate the training of machine readers for the purpose along the lines of the machine-interpretable wet-lab protocols.

3 THE NLPCONTRIBUTIONS MODEL

3.1 Goals

The development of the NLPContributions annotation model was backed by four primary goals:

(1) We aim to produce a semantic representation based on ex-isting work, that can be well motivated as an annotation scheme for the application domain of NLP-ML scholarly ar-ticles, and is specifically aimed at the knowledge capture of the *contributions* in scholarly articles;

(2) The annotated scholarly contributions based on NLPCONTRI-BUTIONS should be integrable in the Open Research Knowledge Graph (ORKG)[1] –the state-of-the-art content-based knowledge capturing platform of scholarly articles' contributions.

(3) The NLPCONTRIBUTIONS model should be useful to produce data for the development of machine learning models in the form of machine readers [18] of scholarly contributions. Such trained models can serve to automatically extract such structured information for downstream applications, either in completely automated or semi-automated workflows as recommenders.[2]

(4) The NLPCONTRIBUTIONS model should be amenable to feedback via a consensus approval or content annotation change suggestions from a large group of authors toward their scholarly article contribution descriptions (an experiment that is beyond the scope of the present work and planned as following work).

The NLPCONTRIBUTIONS annotation model is designed for building a knowledge graph. It is not ontologized, therefore, we assume a bottom-up data-driven design toward ontology discovery as more annotated contributions data is available. Nonetheless, we do propose a core skeleton model for organizing the information at the top-level KG nodes. This involves a root node called CONTRIBUTION, following which, at the first level of the knowledge graph, are ten nodes representing core information units under which the scholarly contributions data is organized.

3.2 The Ten Core Information Units

In this section, we describe the ten information units in our model.

RESEARCHPROBLEM. Determines the research challenge addressed by a contribution using the predicate *hasResearchProblem*. By definition, it is the focus of the research investigation, in other words, *the issue for which the solution must be obtained.*

The task entails identifying only the research problem addressed in the paper and not research problems in general. For instance, in the paper about the BioBERT word embeddings [30], their research problem is just the 'domain-customization of BERT' and not 'biomedical text mining,' since it is a secondary objective.

The RESEARCHPROBLEM is typically found in an article's Title, Abstract and first few paragraphs of the Introduction. The task involves annotating one or more sentences and precisely the research problem phrase boundaries in the sentences.

The subsequent seven information objects are connected to CONTRIBUTION via the generic predicate *has*.

APPROACH. Depending on the paper's content, is referred to as MODEL or METHOD or ARCHITECTURE or SYSTEM or APPLICATION. Essentially, this is the contribution of the paper as *the solution proposed for the research problem.*

The annotations are made only for the high-level overview of the approach without going into system details. Therefore, the equations associated with the model and all the system architecture figures are not part of the annotations. While annotating the earlier

---

[1]https://www.orkg.org/orkg/
[2]In future work, we will expand our current pilot annotated dataset of 50 articles with at least 400 additional similarly annotated articles to facilitate machine learning.

```
{
  "has" : {
    "Approach" : {
      "converting questions" : {
        "to (un-interpretable) vectorial representations" : {
          "which require" : "no pre-defined grammars or lexicons",
          "can query" : {
            "any KB" : {
              "independent of" : "schema"
            }
          }
        }
      },
      "from sentence" : "In this paper, we instead take the
        approach of converting questions to (un-interpretable)
        vectorial representations which require no pre-defined
        grammars or lexicons and can query any KB independent of
        its schema."
    }
  }
}
```

Figure 1: Fine-grained modeling illustration from a single sentence for part of an APPROACH proposed in [9].

RESEARCHPROBLEM did not involve semantic annotation granularity beyond one level, annotating the APPROACH can. Sometimes the annotations (one or multi-layered) are created using the elements within a single sentence itself (see Figure 1); at other times, if they are multi-layered semantic annotations, they are formed by bridging two or more sentences based on their coreference relations. For the annotation element content itself, while, in general, the subject, predicate, and object phrases are obtained directly from the sentence text, at times the predicate phrases have to be introduced as generic terms such as "has" or "on" or "has description" wherein the latter predicate is used for including, as objects, longer text fragments within a finer annotation granularity to describe the top-level node. The actual type of approach is restricted to those sub-types stated in the beginning of the paragraph and is decided based on the the reference to the solution used by the authors or the solution description section name itself. If the reference to the solution or its section name is specific to the paper, such as 'Joint model,' then we rename it to just 'Model.' In general, any alternate namings of the solution, other than those mentioned earlier, including "idea", are normalized to "Model." Finally, as machine learning solutions, they are often given names. E.g., the model BioBERT [30], in which case we introduce the predicate 'called,' as in (Method, called, BioBERT).

The APPROACH is found in the article's Introduction section in the context of cue phrases such as "we take the *approach*," "we propose the *model*," "our system *architecture*," or "the *method* proposed in this paper." However, there are exceptions when the Introduction does not present an overview of the system, in which case we analyze the first few lines within the main system description content in the article. Also, if the paper refers to their system by "method" or "application," this is normalized to APPROACH information unit. SYSTEM or ARCHITECTURE is MODEL information unit.

OBJECTIVE. This is the *defined function for the machine learning algorithm to optimize over.*

In some cases, the APPROACH objective is a complex function. In such cases, it is isolated as a separate information object connected directly to the CONTRIBUTION.

EXPERIMENTALSETUP. Has the alternate name HYPERPARAMETERS. It includes details about the platform including both hardware (e.g., GPU) and software (e.g., Tensorflow library) for implementing the machine learning solution; and of variables, that determine the network structure (e.g., number of hidden units) and how the network is trained (e.g., learning rate), for tuning the software to the task objective.

Recent machine learning models are all neural based and such models have several associated variables such as hidden units, model regularization parameters, learning rate, word embedding dimensions, etc. Thus to offer users a glance at the contributed system, this aspect is included in NLPCONTRIBUTIONS. We only model the experimental setup that are expressed in a few sentences or that are concisely tabulated. There are cases when the experimental setup is not modeled at all within NLPCONTRIBUTIONS. E.g., for the complex "machine translation" models that involve many parameters. Thus, whether the experimental setup should be modeled or not, may appear as a subjective decision, however, over the course of several annotated articles becomes apparent especially when the annotator begins to recognize the simple sentences that describe the experimental setup.

The EXPERIMENTALSETUP unit is found in the sections called Experiment, Experimental Setup, Implementation, Hyperparameters, or Training.

RESULTS. Are the main findings or outcomes reported in the article for the RESEARCHPROBLEM.

Each RESULT unit involves some of the following elements: {dataset, metric, task, performance score}. Regardless of how the sentence(s) are written involving these elements, we assume the following precedence order: [dataset -> task -> metric -> score] or [task -> dataset -> metric -> score], as far as it can be applied without significantly changing the information in the sentence. Consider this illustrated in Figure 2. In the figure, the JSON is arranged starting at the dataset, followed by the task, then the metric, and finally the actual reported result. While this information unit is named per those stated in the earlier paragraph, if in a paper the section name is non-generic, e.g., "Main results," "End-to-end results," it is normalized to a default name "Results."

The RESULTS unit is found in the Results, Experiments, or Tasks sections. While the results are often highlighted in the Introduction, unlike the APPROACH unit, in this case, we annotate the dedicated, detailed section on Results because results constitute a primary aspect of the contribution. Next we discuss the TASKS information unit, and note that RESULTS can include TASKS and vice versa as we describe next.

TASKS. : The APPROACH or MODEL, particularly in multi-task settings, are tested on more than one task, in which case, we list all the experimental tasks. The experimental tasks are often synonymous with the experimental datasets since it is common in NLP for tasks to be defined over datasets. Where lists of TASKS are concerned, the TASKS can include one or more of the EXPERIMENTALSETUP, HYPERPARAMETERS, and RESULTS as sub information units.

```
{
    "CoNLL test set" : {
        "for" : {
            "NER" : {
                "F1-score" : "91.57%"
            }
        },
        "from sentence" : "For NER (Table 7), S-LSTM
            gives an F1-score of 91.57% on the CoNLL
            test set, which is significantly better
            compared with BiLSTMs."
    }
}
```

Figure 2: Illustration of modeling of RESULT (from [53]) w.r.t. a precedence of its elements as [dataset -> task -> metric -> score].

EXPERIMENTS. Are an encompassing information unit that includes one or more of the earlier discussed units. Can include a combination of EXPERIMENTALSETUP and RESULTS, or it can be combination of lists of TASKS and their RESULTS, or a combination of APPROACH, EXPERIMENTALSETUP and RESULTS.

Recently, more and more multitask systems are being developed. Consider, the BERT model [15] as an example. Therefore, modeling EXPERIMENTALSETUP with RESULTS or TASKS with RESULTS is necessary in such systems since the experimental setup often changes per task producing a different set of results. Hence, this information unit encompassing two or more sub information units is relevant.

ABLATIONANALYSIS. Is a form of RESULTS that describes the performance of components in systems.

Unlike RESULTS, ABLATIONANALYSIS is not performed in all papers. Further, in papers that have them, we only model these results if they are expressed in a few sentences, similar to our modeling decision for HYPERPARAMETERS.

The ABLATIONANALYSIS information unit is found in the sections that have Ablation in their title. Otherwise, it can also be found in the written text without having a dedicated section for it. For instance, in the paper "End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures" [35] there is no section title with Ablation, but this information is extracted from the text via cue phrases that indicate ablation results are being discussed.

BASELINES. are those listed systems that a proposed approach is compared against.

The BASELINES information unit is found in sections that have Baseline in their title. Otherwise, it can also be found in sections that are not directly titled Baseline, but require annotator judgement to infer that baseline systems are being discussed. For instance, in the paper "Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers," [50] the baselines are discussed in subsection 'Methods.' Or in paper "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," [41], the baselines are discussed in a section called "Previous State-of-the-Art."

Of these ten information units, only three are mandatory. They are Research Problem, Approach, and Results; the other seven may or may not be present depending on the content of the article.

Code. is a link to the software on Github or on other similar open source platforms, or even on author's website.

3.3 Contribution Sequences within Information Units

Except for Research Problem, each of the remaining nine information units encapsulate different aspects of the contributions of scholarly investigations in the NLP-ML domain; with the Research Problem offering the primary contribution context. Within the seven different aspects, there are what we call Contribution Sequences.

Here, with the help of an example depicted in Figure 3 we illustrate the notion of contribution sequences. In this example, we model contribution sequences in the context of the Experimental Setup information unit. In the figure, this information unit has two contribution sequences. The first connected by predicate 'used' to the object 'BERTBase model,' and the second, also connected by predicate 'used' to the object 'NVIDIA V100 (32GB) GPUs.' The 'BERTBase model' contribution sequence includes a second level of detail expressed via two different predicates 'pre-trained for' and 'pre-trained on.' As a model of scientific knowledge, the triple with the entities connected by the first predicate, i.e. (BERTBase model, pre-trained for, 1M steps) reflects that the 'BertBase model' was pretrained for 1 million steps. The second predicate produces two triples: (BERTBase model, pre-trained on, English Wikipedia) and (BERTBase model, pre-trained on, BooksCorpus). In each case, the scientific knowledge captured by these two triples is that BERTBase was pretrained on {Wikipedia, BooksCorpus}. Note in the JSON data structure, the predicate connects the two objects as an array. Next, the second contribution sequence, hinged at 'NVIDIA V100 (32GB) GPUs' as the subject has two levels of granularity. Consider the following three triples: (NVIDIA V100 (32GB) GPUs, used, ten) and (ten, for, pre-training). Note, in this nesting pattern, except for 'NVIDIA V100 (32 GB) GPUs,' the predicates {used, for} and remaining entities {ten, pre-training} are nested according to their order of appearance in the written text. Therefore, in conclusion, an information unit can have several contribution sequences, and the contribution sequences need not be identically modeled. For instance, our second contribution sequence is modeled in a fine grained manner, i.e. in multiple levels. And when fine-grained modeling is employed, it is relatively straightforward to spot in the sentence(s) being modeled.

4 THE PILOT ANNOTATION TASK

The pilot annotation task was performed by a postdoctoral researcher with a background in natural language processing. The NLPContributions model or scheme just described, were developed over the course of the pilot task. At a high-level, the annotations were performed in three main steps. They are presented next, after which we describe the annotation guidelines.

```
{
  "has" : {
    "Experimental setup" : {
      "used" : [
        {
          "BERTBase model" : {
            "pre-trained for" : "1M steps",
            "pre-trained on" : ["English Wikipedia",
              "BooksCorpus"]
          },
          "from sentence" : "We used the BERTBASE model
            pre-trained on English Wikipedia and
            BooksCorpus for 1M steps."
        },
        {
          "NVIDIA V100 (32GB) GPUs" : {
            "used" : {
              "eight" : {
                "for" : "pre-training"
              }
            },
            "from sentence" : "We used eight NVIDIA V100
              (32GB) GPUs for the pre-training."
          }
        }
      ]
    }
  }
}
```

Figure 3: Illustration of the modeling of Contribution Sequences in the Experimental Setup Information Unit (from [30]). Created using https://jsoneditoronline.org

4.1 Pilot Task Steps

(a) Contribution-Focused Sentence Annotations. In this stage, sentences from scholarly articles were selected as candidate contribution sentences under each of the aforementioned mandatory three information units (viz., Research Problem, Approach, and Results) and, if applicable to the article, for one or more of the remaining seven information units as well.

To identify the contribution sentences in the article, the full-text of the article is searched. However, as discussed at the end of Section 2, the Background, Related Work, and Conclusions sections are entirely omitted from the search. Further, the section discussing the Approach or the System is only referred to when the Introduction section does not offer sufficient highlights of this information unit. In addition, except for tabulated hyperparameters, we do not consider other tables for annotation within the NLPContributions model.

To better clarify the pilot task process, in this subsection, we use Figure 2 as the running example. From the example, at this stage, the sentence "For NER (Table 7), S-LSTM gives an F1-score of 91.57% on the CoNLL test set, which is significantly better compared with BiLSTMs." is selected as one of the contribution sentence candidates as part of the Results information unit. This sentence is selected from a Results subsection in [53], but is just one among three others.

(b) Chunking Phrase Spans for Subject, Predicate, Object Entities. Then for the selected sentences, we annotate their scientific knowledge entities. The entities are annotated by annotators having an implicit understanding of whether they take the subject, predicate, or object roles in a per triple context. As a note, by our annotation scheme, predicates are not mandatorily verbs and can be nouns as well.

Resorting to our running example, for the selected sentence, this stage involves annotating the phrases "For," "NER," "F1-score," "91.57%," and "CoNLL test set," with the annotator cognizant of the fact that they will use the [dataset -> task -> metric -> score] scientific entity precedence in the next step.

(c) Creating contribution sequences. This involves relating the subjects and objects within triples, which as illustrated in Section 3.3, the object in one triple can be a subject in another triple if the annotation is performed at a fine-grained level of detail. For the most part, the nesting is done per order of appearance of the entities in the text, except for those involving the scientific entities {dataset, task, metric, score} under the RESULTS information unit.

In the context of our running example, given the early annotated scientific entities, in this stage, the annotator will form the following two triples: (CoNLL test set, For, NER), (NER, F1-score, 91.57%) as a single contribution sequence. What is not depicted in Figure 1 are the top-level annotations including the root node and one of the ten information unit nodes. This is modeled as follows: (Contribution, has, Results), and (Results, has, CoNLL test set).

4.2　Task Guidelines

In this section, we elicit a set of general guidelines that inform the annotation task.

How are information unit names selected? For information units such as APPROACH, EXPERIMENTALSETUP, and RESULTS that each have a set of candidate names, the applied name is the one selected based on the closest section title or cue phrase.

Which of the ten information units does the sentence belong to? Conversely to the above, if a sentence is first identified as a contribution sentence candidate, it is placed within the information unit category that is identified directly based on the section header for the sentence in the paper or inferred from cue phrases from the first few sentences in its section.

Inferring Predicates. In ideal settings, the constraint on the text used for subjects, objects, and predicates in contribution sequences is that they should be found in their corresponding sentence. However, for predicates this is not always possible. Since predicate information may not always be found in the text, it is sometimes annotated additionally based on the annotator judgment. However, even this open-ended choice remains restricted to a predefined set of candidates. It includes {"has", "on", "by", "for", "has value", "has description", "based on", "called"}.

How are the supporting sentences linked to their corresponding contribution sequence within the overall JSON object? The sentence(s) is stored in a dictionary with a "from sentence" key, which is then attached to either the first element or, if it is a nested triples hierarchy, sometimes even to the second element of a contribution sequence. The dictionary data-type containing the evidence

sentence is either put as an array element, or as a nested dictionary element.

Are the nested contribution sequences always obtained from a single sentence? The triples can be nested based on information from one or more sentences in the article. Further, the sentences need not be consecutive in the running text. As mentioned earlier, the evidence sentences are attached to the first element or the second element by the predicate "from sentence." If a contribution sequence is generated from a table then the table number in the original paper is referenced.

When is the Approach actually modeled from the dedicated section as opposed to the Introduction? In general, we avoid annotating the Approach or Model sections for their contribution sentences as they tend to delve deeply into the approach or model details, and involve complicated elements such as equations, etc. Instead, we restrict ourselves to the system higlights in the Introduction. However, in some articles the Introduction doesn't offer system highlights which is when we resort to using the dedicated section for the contribution highlights in this mandatory information unit.

Do we explore details about hardware used as part of the contribution? Yes, if it is explicitly part of the hyperparameters.

Are predicates always verbs? Predicates are not always verbs. They can also be nouns especially in the hyperparameters section.

Creating contribution sequences from tabulated hyperparameters. Only for hyperparameters, we model their tabulated version if given. This is done as follows: 1) for the predicate, we use the name of the parameter; and 2) for the object, the value against the name. Sometimes, however, if there are two-level hierarchical parameters, then the predicate is the first name, object is the value, and the value is qualified by the parameter name lower in the hierarchy. Qualifying the second name involves introducing the "for" predicate.

How are lists modeled within contribution sequences? As part of the contribution sentence candidates, are also included sentences with lists. Such sentences are predominantly found for the EXPERIMENTALSETUP or RESULT information units. This is modeled as depicted in Figure 4 for the first two list elements. Here, the MODEL information unit has two contribution sequences, each pertaining to a specific list item in the sentence. Further, the predicate "has description" is introduced for linking text descriptions.

Which JSON structures are used to represent the data? Flexibly, they include dictionaries, or nested dictionaries, or arrays of items, where the items can be strings, dictionaries, nested dictionaries, or arrays themselves.

How are appositives handled? We introduce a new predicate "name" to handle appositives.

5　MATERIALS AND TOOLS

5.1　Paper Selection

A collection of scholarly articles is downloaded based on the ones in the publicly available leaderboard of tasks in artificial intelligence called https://paperswithcode.com/. It predominantly represents papers in the Natural Language Processing and Computer Vision fields. For the purposes of our NLPCONTRIBUTIONS model, we restrict ourselves just to the NLP papers. From the set, we randomly

```
{
  "has" : {
    "Model" : {
      "consists of" : {
        "Word Representation" : {
          "has description" : "maps each word in a
             sentence into vector representations"
        },
        "Self Attention" : {
          "has description" : "captures the meaning of
             the correlation between words based on multi
             -head attention"
        },
        "from sentence" : "As shown in Figure 2, our
           model consists of four main components: (1)
           Word Representation that maps each word in a
           sentence into vector representations; (2) Self
           Attention that captures the meaning of the
           correlation between words based on multi-head
           attention [20];"
      }
    }
  }
}
```

Figure 4: Illustration of the modeling of a sentence with a list as part of the MODEL Information Unit (from [29]). Created using https://jsoneditoronline.org

select 10 papers in five different NLP-ML research tasks: 1. machine translation, 2. named entity recognition, 3. question answering, 4. relation classification, and 5. text classification.
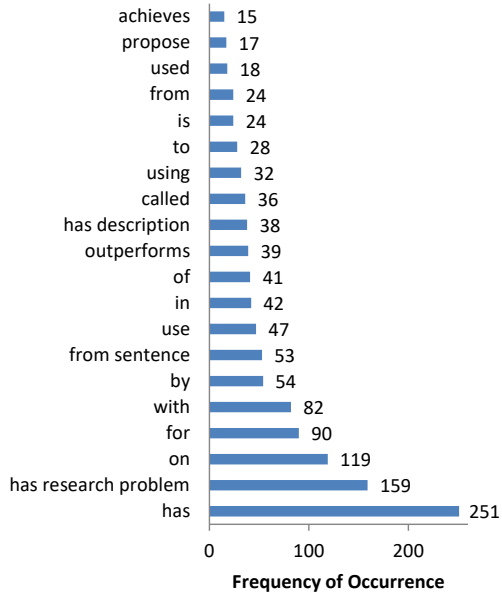
Figure 5: A list of the predicates in our triples dataset that appear more than 15 times.

5.2 Data Representation Format and Annotation Tools

JSON was the chosen data format for storing the semantified parts of the scholarly articles contributions. To avoid syntax errors in creating the JSON objects, the annotations were made via https://jsoneditoronline.org which imposes valid JSON syntax checks. Finally, in the early stages of the annotation task, some of the annotations were made manually in the ORKG infrastructure https://www.orkg.org/orkg/ to test their practical suitability in a knowledge graph; three of such annotated papers are depicted in Figure 6. The links in the Figure captions can be visited to explore the annotations at their finer granularity of detail.

5.3 Annotated Dataset Characteristics

Overall, the annotated corpus contains a total of 2631 triples (avg. of 52 triples per article). Its data elements comprise 1033 unique subjects, 843 unique predicates, and 2182 unique objects. In Table 1 below, we show the per-task distribution of triples and their elements. Of all tasks, relation classification has the highest number of unique triples (544) and named entity recognition the least (473).

Generally, in the context of triples formation, predicates are often selected from a closed set and hence comprise a smaller group of items. In the NLPCONTRIBUTION model, however, predicates are extracted from the text if present. This leads to a much larger set of predicates that would require the application of predicate normalization functions to find the smaller core semantic set. In Figure 5, to offer some insights to this end, we show the predicates that appear more than 15 times over all the triples. We find the predicate *has* appears most frequently since its function often serves as a filler predicate. A complete list of the predicates is released in our dataset repository online https://doi.org/10.25835/0019761.

	MT	NER	QA	RC	TC
Subject	259	209	203	228	221
Predicate	243	220	187	201	252
Object	471	434	515	455	459
Total	502	473	497	544	504

Table 1: Per-task (machine translation (MT), named entity recognition (NER), question answering (QA), relation classification (RC), text classification (TC)) triples distribution in terms of unique subject, predicate, object, and overall.

6 USE CASE: NLPCONTRIBUTIONS IN ORKG

As a use case of the ORKG infrastructure, instead of presenting just the annotations obtained from NLPCONTRIBUTIONS, we present a further enriched showcase. Specifically, we model the evolution of the annotation scheme at three different attempts with the third one arriving at NLPCONTRIBUTIONS. This is depicted in Figure 6. Our use case is an enriched one for two reasons: 1) it depicts the ORKG infrastructure flexibility for data-driven ontology discovery that makes allowances for different design decisions; and 2) it also shows how within flexible infrastructures the possibilities can be too wide that arriving at a consensus can potentially prove a challenge if it isn't mandated at a critical point in the data accumulation.

Research problems

Add to comparison

different architectures for the relation encoder ftheta, all built on top of the widely used
Transformer sequence model (Devlin et al., 2018; Vaswani et al., 2017)

Contribution data

Has	results
Introduce	three separate methods of extracting a fixed length relation representation hr from the BERT encoder
Perform	task-specific fine-tuning of the BERT model
Present	three different options for getting information about the focus spans s1 and s2 into our BERT encoder

(a) Research paper [45] top-level snapshot in ORKG https://www.orkg.org/orkg/paper/R41467/

Research problems

Add to comparison

how to effectively make use of relevant information while ignoring irrelevant information from the
dependency trees

Contribution data

Code	https://github.com/Cartus/AGGCN_TACRED 🗗
Has	Attention Guided GCNs
	Baselines
	binary-class n-ary relation extraction
	Experiments
	multi-class classification task
	results
Propose	novel Attention Guided Graph Convolutional Networks (AGGCNs)

(b) Research paper [21] top-level snapshot in ORKG https://www.orkg.org/orkg/paper/R41374

Research problems

Add to comparison

extension of graph convolutional networks that is tailored for relation extraction, which pools
information over arbitrary dependency structures efficiently in parallel
Relation extraction

Contribution data

Has	Ablation study
	Baseline Models
	Model
	Results

(c) Research paper [54] top-level snapshot in ORKG https://www.orkg.org/orkg/paper/R44287

Figure 6: Figures 6(a),6(b),6(c) depict evolution of the annotation scheme over three different research papers. Fig. 6(c) is the resulting selected format NLPCONTRIBUTIONS that is proposed in this paper.

Figure 6(a) depicts the first modeling attempts of an NLP-ML contribution. For predicates, the model restricts itself to use only those found in the text. The limitation of such a model is that not normalizing linguistic variations very rarely creates comparable models across investigations even if they imply the same thing. Hence, we found that for comparability a common predicate vocabulary at the top-level in the model minimally needs to be in place. Figure 6(b) is the second attempt of modeling a different NLP-ML contribution. In this attempt, the predicates at the top-level are mostly normalized to a generic "has," however, "has" is connected to various information items again lexically based on the text of the scholarly articles, one or more of which can be grouped under a common category. Via such observations, we systematized the knowledge organization at the top-level of the graph by introducing the ten information unit nodes. Figure 6(c) is the resulting NLP-CONTRIBUTIONS annotations model. Within this model, scholarly contributions with one or more of the information units in common, viz. "Ablation study," "Baseline Models," "Model," and "Results," can be uniformly compared.

7 LIMITATIONS

Obtaining disjoint (subject, predicate, object) triples as contribution sequences. It was not possible to extract disjoint triples from all sentences. In many cases, we extract the main predicate and use as object the relevant full sentence or its clausal part. From [30], for instance, under the EXPERIMENTALRESULTS information unit, we model the following: (Contribution, has, Experimental results); (Experimental results, on, all datasets); and (all datasets, achieves, BioBERT achieves higher scores than BERT). Note, in the last triple, "achieves" was used as a predicate and its object "BioBERT achieves higher scores than BERT" is modeled as a clausal sentence part.

Employing coreference relations between scientific entities. In the fine-grained modeling of schemas, scientific entities within triples are sometimes nested across sentences by leveraging their coreference relations. We consider this a limitation toward the automated machine reading task, since coreference resolution itself is often challenging to perform automatically.

Tabulated results are not incorporated within NLPCONTRIBU-TIONS. Unlike tabulated hyperparameters which have a standard format, tabulated results have significantly varying formats. Thus their automated table parsing is a challenging task in itself. Nonetheless, by considering the textual results, we relegate ourselves to their summarized description, which often serves sufficient for highlighting the contribution.

Can all NLP-ML papers be modeled by NLPCONTRIBUTIONS? While we can conclude that some papers are easier to model than others (e.g., articles addressing 'relation extraction' vs. 'machine translation' which are harder), it is possible that all papers can be modelled by at least some if not all the information units of the model we propose.

8 DISCUSSION

From the pilot dataset annotation exercise, we note the following regarding task practically. Knowledge modeled under some information units are more amenable to systematic structuring than others. E.g., information units such as RESEARCHPROBLEM, EXPERI-MENTALSETUP, RESULTS, and BASELINES are readily amenable for systematic templates discovery toward their structured modeling within the ORKG; whereas the remaining information units, especially APPROACH or MODEL, will require additional normalization steps toward the search for their better structuring.

9 CONCLUSIONS AND FUTURE DIRECTIONS

The Open Research Knowledge Graph [3] makes scholarly knowledge about research contributions machine-actionable: i.e. findable, structured, and comparable. Manually building such a knowledge graph is time-consuming and requires the expertise of paper authors and domain experts. In order to efficiently build a scholarly knowledge contributions graph, we will leverage the technology of machine readers [18] to assist the user in annotating scholarly article contributions. But the machine readers will need to be trained for such a task objective. To this end, in this work, we have proposed an annotation scheme for capturing the contributions in natural language processing scholarly articles, in order to create such training datasets for machine readers. In addition, we also provide a set of 50 annotated articles by the NLPCONTRIBUTIONS scheme as a practical demonstration of feasibility of the annotation task. However, for the training of machine learning models in future work we will release a larger dataset annotated by the proposed scheme. To facilitate future research, our pilot dataset is released online at https://doi.org/10.25835/0019761.

Finally, aligned with the initiatives within research communities to build the Internet of FAIR Data and Services (IFDS) [6], the data within ORKG are compliant [38] with such FAIR data principles [52] thus making them Findable, Accessible, Interoperable and Reusable. Since the dataset we annotate by our proposed scheme is designed to be ORKG-compliant, we adopt the cutting-edge standard of data creation within the research community.

Nevertheless, the NLPCONTRIBUTION model is a surface semantic structuring scheme for the contributions in unstructured text. To realize a full-fledged machine-actionable and inferenceable knowledge graph of scholarly contributions, as future directions, there are a few IE modules that would need to be improved or added. They are (1) improving the PDF parser to produce less noisy output; (2) incorporating an entity and relation linking and normalization module; (3) merging phrases from the unstructured text with known ontologies (e.g., the MEX vocabulary [17]) to align resources and thus ensure data interoperability and reusability; and (4) extending the model to more scholarly disciplines and domains.

REFERENCES

[1] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. 2018. Construction of the Literature Graph in Semantic Scholar. In *NAACL, Volume 3 (Industry Papers)*. 84–91.

[2] Waleed Ammar, Matthew E. Peters, Chandra Bhagavatula, and Russell Power. 2017. The AI2 system at SemEval-2017 Task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In *SemEval@ACL*.

[3] Sören Auer. 2018. Towards an Open Research Knowledge Graph. https://doi.org/10.5281/zenodo.1157185

[4] Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria Esther Vidal. 2018. Towards a knowledge graph for science. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*. 1–6.

[5] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. In *SemEval@ACL*.

[6] Paul Ayris, Jean-Yves Berthou, Rachel Bruce, Stefanie Lindstaedt, Anna Monreale, Barend Mons, Yasuhiro Murayama, Caj Södergård, Klaus Tochtermann, and Ross Wilkinson. 2016. *Realising the European open science cloud*. Luxembourg. https://doi.org/10.2777/940154

[7] Niranjan Balasubramanian, Stephen Soderland, Oren Etzioni, et al. 2013. Generating coherent event schemas at scale. In *EMNLP*. 1721–1731.

[8] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *EMNLP-IJCNLP*. 3606–3611.

[9] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Joint European conference on machine learning and knowledge discovery in databases*. 165–180.

[10] Arthur Brack, Jennifer D'Souza, Anett Hoppe, Sören Auer, and Ralph Ewerth. 2020. Domain-Independent Extraction of Scientific Concepts from Research Articles. In *Advances in Information Retrieval*. Springer International Publishing, 251–266.

[11] Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *EMNLP*. 1797–1807.

[12] Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*. 789–797.

[13] Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. 602–610.

[14] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. 2016. The document components ontology (DoCO). *Semantic web* 7, 2 (2016), 167–181.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota, 4171–4186.

[16] Jennifer D'Souza, Anett Hoppe, Arthur Brack, Mohmad Yaser Jaradeh, Sören Auer, and Ralph Ewerth. 2020. The STEM-ECR Dataset: Grounding Scientific Entity References in STEM Scholarly Content to Authoritative Encyclopedic and Lexicographic Sources. In *LREC*. Marseille, France, 2192–2203.

[17] Diego Esteves, Diego Moussallem, Ciro Baron Neto, Tommaso Soru, Ricardo Usbeck, Markus Ackermann, and Jens Lehmann. 2015. MEX vocabulary: a lightweight interchange format for machine learning experiments. In *Proceedings of the 11th International Conference on Semantic Systems*. 169–176.

[18] Oren Etzioni, Michele Banko, and Michael J Cafarella. 2006. Machine Reading.. In *AAAI*, Vol. 6. 1517–1519.

[19] Said Fathalla, Sahar Vahdati, Sören Auer, and Christoph Lange. 2017. Towards a knowledge graph representing research findings by semantifying survey articles. In *TPDL*. Springer, 315–327.

[20] Beatríz Fisas, Francesco Ronzano, and Horacio Saggion. 2016. A Multi-Layered Annotated Corpus of Scientific Papers. In *LREC*.

[21] Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *ACL*. 241–251.

[22] Siegfried Handschuh and Behrang QasemiZadeh. 2014. The ACL RD-TEC: a dataset for benchmarking terminology extraction and classification in computational linguistics. In *COLING 2014: 4th international workshop on computational terminology*.

[23] John PA Ioannidis. 2016. The mass production of redundant, misleading, and conflicted systematic reviews and meta-analyses. *The Milbank Quarterly* 94, 3 (2016), 485–514.

[24] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge. In *KCAP* (Marina Del Rey, CA, USA). ACM, New York, NY, USA, 243–246.

[25] Arif E Jinha. 2010. Article 50 million: an estimate of the number of scholarly articles in existence. *Learned Publishing* 23, 3 (2010), 258–263.

[26] Olga Kononova, Haoyan Huo, Tanjin He, Ziqin Rong, Tiago Botari, Wenhao Sun, Vahe Tshitoyan, and Gerbrand Ceder. 2019. Text-mined dataset of inorganic materials synthesis recipes. *Scientific data* 6, 1 (2019), 1–11.

[27] Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. 2018. An Annotated Corpus for Machine Reading of Instructions in Wet Lab Protocols. In *NAACL: HLT, Volume 2 (Short Papers)*. New Orleans, Louisiana, 97–106. https://doi.org/10.18653/v1/N18-2016

[28] Fusataka Kuniyoshi, Kohei Makino, Jun Ozawa, and Makoto Miwa. 2020. Annotating and Extracting Synthesis Process of All-Solid-State Batteries from Scientific Literature. In *LREC*. 1941–1950.

[29] Joohong Lee, Sangwoo Seo, and Yong Suk Choi. 2019. Semantic Relation Classification via Bidirectional LSTM Networks with Entity-aware Attention using Latent Entity Typing. *arXiv preprint arXiv:1901.08163* (2019).

[30] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[31] Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. 2012. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics* 28, 7 (2012), 991–1000.

[32] Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin R. Batchelor. 2010. Corpora for the Conceptualisation and Zoning of Scientific Papers. In *LREC*.

[33] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *EMNLP*.

[34] Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. *arXiv preprint arXiv:1708.06075* (2017).

[35] Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th ACL (Volume 1: Long Papers)*. 1105–1116.

[36] Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The Materials Science Procedural Text Corpus: Annotating Materials Synthesis Procedures with Shallow Semantic Structures. In *Proceedings of the 13th Linguistic Annotation Workshop*. 56–64.

[37] Allard Oelen, Mohamad Yaser Jaradeh, Kheir Eddine Farfar, Markus Stocker, and Sören Auer. 2019. Comparing Research Contributions in a Scholarly Knowledge Graph. In *K-CAP 2019*. 21–26.

[38] A. Oelen, M. Y. Jaradeh, M. Stocker, and S. Auer. 2020. Generate FAIR Literature Surveys with Scholarly Knowledge Graphs.

[39] Vayianos Pertsas and Panos Constantopoulos. 2017. Scholarly Ontology: modelling scholarly practices. *International Journal on Digital Libraries* 18, 3 (2017), 173–190.

[40] Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals and understanding: An inquiry into human knowledge structures. (1977).

[41] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).

[42] Dan Simonson and Anthony Davis. 2015. Interactions between narrative schemas and document categories. In *Proceedings of the First Workshop on Computing News Storylines*. 1–10.

[43] Dan Simonson and Anthony Davis. 2016. NASTEA: Investigating narrative schemas through annotated entities. In *Proceedings of the 2nd Workshop on Computing News Storylines*. 57–66.

[44] Dan Simonson and Anthony Davis. 2018. Narrative Schema Stability in News Text. In *Proceedings of the 27th International Conference on Computational Linguistics*. 3670–3680.

[45] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *ACL*. 2895–2905.

[46] Larisa N. Soldatova and Ross D. King. 2006. An ontology of scientific experiments. *Journal of the Royal Society, Interface* 3 11 (2006), 795–803.

[47] Simone Teufel, Jean Carletta, and Marc Moens. 1999. An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the ninth conference on European chapter of ACL*. 110–117.

[48] Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 2009. Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *EMNLP: Volume 3*. 1493–1502.

[49] Lars Vogt, Jennifer D'Souza, Markus Stocker, and Sören Auer. 2020. Toward Representing Research Contributions in Scholarly Knowledge Graphs Using Knowledge Graph Cells. In *JCDL '20, August 1–5, 2020, Virtual Event, China*.

[50] Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers. In *Proceedings of the 57th ACL*. 1371–1377.

[51] Mark Ware and Michael Mabe. 2015. The STM Report: An overview of scientific and scholarly journal publishing. (03 2015).

[52] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016).

[53] Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-State LSTM for Text Representation. In *Proceedings of the 56th ACL (Volume 1: Long Papers)*. 317–327.

[54] Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *EMNLP*. 2205–2215.

A   TWICE MODELING AGREEMENT

In general, even if the annotations are performed by a single anno-tator, there will be an annotation discrepancy. Compare the same information unit "Experimental Setup" modeled in Figure 7 below versus that modeled in Figure 3. Fig. 7 was the first annotation at-tempt and includes the second attempted model, done on a different day and blind from the the first. While neither are incorrect, the sec-ond has taken the least annotated information route possibly due to annotator fatigue, hence a two-pass methodology is recommended.

```json
{
  "has" : {
    "Experimental Setup" : {
      "BioBERT v1.0 (+ PubMed + PMC)" : {
        "trained for" : "470K steps",
        "from sentence" : "BioBERT v1.0 (+ PubMed + PMC) is the version of BioBERT (+ PubMed + PMC) trained for 470K steps."
      },
      "PubMed" : {
        "found" : {
          "200K" : {
            "optimal" : "pre-training steps"
          }
        },
        "from sentence" : "When using both the PubMed and PMC corpora, we found that 200K and 270K pre-training steps were optimal for PubMed and PMC, respectively."
      },
      "PMC" : {
        "found" : {
          "270K" : {
            "optimal" : "pre-training steps"
          }
        },
        "from sentence" : "When using both the PubMed and PMC corpora, we found that 200K and 270K pre-training steps were optimal for PubMed and PMC, respectively."
      },
      "batch size and learning rate scheduling" : {
        "same as those" : "pre-training BERT",
        "from sentence" : "Other hyper-parameters such as batch size and learning rate scheduling for pre-training BioBERT are the same as those for pre-training BERT unless stated otherwise."
      },
      "Naver Smart Machine Learning (NSML) (Sung et al., 2017" : {
        "using" : "pre-trained BioBERT",
        "from sentence" : "We pre-trained BioBERT using Naver Smart Machine Learning (NSML) (Sung et al., 2017), which is utilized for large-scale experiments that need to be run on several GPUs."
      },
      "NVIDIA V100 (32GB) GPUs" : {
        "used" : {
          "eight" : {
            "for" : "pre-training"
          }
        },
        "from sentence" : "We used eight NVIDIA V100 (32GB) GPUs for the pre-training."
      },
      "maximum sequence length" : {
        "fixed" : "512",
        "from sentence" : "The maximum sequence length was fixed to 512 and the mini-batch size was set to 192, resulting in 98 304 words per iteration."
      },
      "mini-batch size" : {
        "set to" : "192",
        "from sentence" : "The maximum sequence length was fixed to 512 and the mini-batch size was set to 192, resulting in 98 304 words per iteration."
      },
      "NVIDIA Titan Xp (12GB) GPU" : {
        "used" : {
          "single" : {
            "to" : "fine-tune BioBERT"
          }
        },
        "from sentence" : "We used a single NVIDIA Titan Xp (12GB) GPU to fine-tune BioBERT on each task."
      },
      "fine-tuning" : {
        "batch size" : "10, 16, 32, or 64",
        "learning rate" : "5e-5, 3e-5 or 1e-5",
        "from sentence" : "For finetuning, a batch size of 10, 16, 32 or 64 was selected, and a learning rate of 5e-5, 3e-5 or 1e-5 was selected."
      }
    }
  }
}
```

Figure 7: Illustration of modeling of Contribution Sequences in the EXPERIMENTAL SETUP Information Unit (from [30]) in a first annotation attempt. Contrast with second attempt depicted in Figure 3 in the main paper content.