Exploiting Unary Relations with Stacked Learning for Relation Extraction

Yuan Zhuang[1], **Ellen Riloff**[1], **Kiri L. Wagstaff**[2],
Raymond Francis[2], **Matthew Golombek**[2], **Leslie Tamppari**[2]

[1]School of Computing, Univeristy of Utah
[2]Jet Propulsion Laboratory, California Institute of Technology
{yyzhuang, riloff}@cs.utah.edu
wkiri@wkiri.com
{raymond.francis, matthew.p.golombek, leslie.k.tamppari}@jpl.nasa.gov

Abstract

Relation extraction models typically cast the problem of determining whether there is a relation between a pair of entities as a single decision. However, these models can struggle with long or complex language constructions in which two entities are not directly linked, as is often the case in scientific publications. We propose a novel approach that decomposes a binary relation into two unary relations that capture each argument's role in the relation separately. We create a stacked learning model that incorporates information from unary and binary relation extractors to determine whether a relation holds between two entities. We present experimental results showing that this approach outperforms several competitive relation extractors on a new corpus of planetary science publications as well as a benchmark dataset in the biology domain.

1 Introduction

For many scientific domains, information extraction (IE) systems can play a valuable role in harvesting information from the scientific literature to automatically populate knowledge bases. Our work is motivated by the goal of populating a Mars knowledge base by extracting information from the planetary science literature about observations made by the rovers on Mars. Specifically, we seek to extract information about the composition and properties of named "Targets" (rocks, soils, dunes, etc.) on the surface of Mars. Figure 1 shows an example sentence from this domain.

> *Several hypotheses could explain the abundance of potassium feldspar observed by CheMin X-ray diffraction of the Windjana drill sample.*

Figure 1: Example sentence from the planetary science domain with a CONTAINS relation between Target *Windjana* and Component *potassium feldspar*.

Our IE task requires identification of three types of entities (Components, Properties, and Targets) and two types of relations (CONTAINS and HASPROPERTY). The Component entities can be minerals or elements. The sentence in Figure 1 mentions one Target (*Windjana*) and one Component (*potassium feldspar*), which participate in a CONTAINS relation. Intuitively, this relation means that potassium feldspar was detected at the Windjana site on Mars [1].

Typically, relation extraction (RE) systems determine whether a pair of entities participate in a relation. In many scientific domains, relation extraction can be challenging because of complex language constructions that do not directly link two relevant entities, even when they occur in the same sentence. For example, the relation in Figure 1 derives from the following complex path: *potassium feldspar was observed by X-ray diffraction ... diffraction of a drill sample ... a drill sample taken at the Windjana site*. This type of sentence structure is challenging for NLP systems to recognize, both lexically and syntactically.

However, even in long or complex sentences, our intuition is that local context is often sufficient to recognize *one* argument of a relation, even when recognizing both arguments simultaneously is difficult. To explore this hypothesis, our research decomposes two-argument (binary) relations into a pair of one-argument *unary* relations and trains separate unary relation extractors for each argument. For example, let us revisit Figure 1 and the CONTAINS relation. We can decompose the binary relation CONTAINS(X,Y) into two unary relations: CONTAINER(X) and CONTAINEE(Y). In Figure 1, the phrase *potassium feldspar observed* strongly suggests the unary relation CONTAINEE(*potassium feldspar*) (i.e., potassium feldspar is part of the

---

[1]The sentence refers to the result of X-ray diffraction by the CheMin instrument (on the Mars Science Laboratory rover) applied to a drill sample at the Windjana site.

composition of something). The phrase *Windjana drill sample* suggests the unary relation CONTAINER(*Windjana*) (i.e., Windjana was studied (drilled) for its composition).

When the local context around one argument is compelling, the unary relation extractor can provide a strong signal that the full binary relation may also exists. But challenges remain with unary relation extractors alone: (1) only one argument may be recognized, and (2) it can be challenging to pair up the individual unary relations correctly. Consequently, we expect that unary relations can be most useful when considered alongside other features to accurately extract binary relations.

In this paper, we present a stacked learning architecture for relation extraction that uses a traditional binary relation extraction model alongside new information from unary relation extractors and features about the entity pair (Section 4). In our stacked learning framework, a meta-classifier makes a decision about a pair of entities based on two perspectives of the sentence context: the broad perspective of the binary relation extraction model and the local perspectives of the corresponding unary relation extractors. As a result, the meta-classifier can be more robust then either approach on its own. We evaluate this stacked learning model on relation extraction tasks for two scientific domains: the Mars mission planetary science domain and a biology domain (chemical-protein interactions) (Section 5). We find that our stacked learning model consistently outperforms traditional binary relation extraction models in both domains.

2 Related Work

Many relation extraction models have used feature-based or kernel-based approaches, such as (Zelenko et al., 2003; Bunescu and Mooney, 2005; Nguyen et al., 2015). Recent relation extraction models often use deep learning methods to learn representations of entities and their contexts and avoid the need for manual feature engineering (e.g., Socher et al., 2012; Zhang and Wang, 2015; Verga et al., 2018; Wang et al., 2019; Christopoulou et al., 2018; Zhang et al., 2018). Many of these methods also fine-tune pretrained language models to better capture contextual information. Such models include BERT (Devlin et al., 2019), SciBERT (Beltagy et al., 2019), which is pretrained over scientific publications, and LinkBERT (Yasunaga et al., 2022), which is pretrained to also capture dependencies between documents.

Pipeline architectures have been widely used for relation extraction, which perform entity recognition as the first stage and then extract relations among the detected entities (e.g., Kambhatla, 2004; Chan and Roth, 2011; Zhong and Chen, 2021). Another approach is to perform entity recognition and relation extraction jointly, which aims to eliminate the problem of error propagation that can occur with pipelines (e.g., Miwa and Bansal, 2016; Zhang et al., 2017; Luan et al., 2019; Wadden et al., 2019; Dixit and Al-Onaizan, 2019; Lin et al., 2020).

Nearly all previous systems make decisions about a relation based on all arguments at the same time. One exception that bears some similarity to our work is (Wei et al., 2020), which trains a classifier to recognize the first argument ("subject") of a relation before trying to detect its second argument ("object"). However, their classifier uses information about the subject to identify the object, and then uses both arguments to make its final decision. In contrast, our unary models completely decouple the tasks of recognizing the first and second arguments of a binary relation.

There has been growing interest in information extraction from scientific publications across a variety of domains (e.g., Gupta and Manning, 2011; Tsai et al., 2013; Tateisi et al., 2014; Li et al., 2016a, 2017; Verga et al., 2018; Watanabe et al., 2019). However, relatively little work has been done for planetary science. The GeoDeepDive project extracts information about rock formations and stratigraphy on Earth from geology publications (Zhang et al., 2013). The Mars Target Encyclopedia project (Wagstaff et al., 2018) extracts named entities (targets, minerals, and elements) and compositional relations from planetary science publications. Their relation extraction component used jSRE (Giuliano et al., 2006), an SVM classifier based on shallow parsing features. We included their data (covering one Mars mission and one relation, CONTAINS) in our experiments, and we augmented it with three more missions, hundreds of additional documents, and a new relation, HASPROPERTY (see Section 5.1). We compare the performance of jSRE models with our relation extraction models in Section 5.4.

3 Mars Target Relations

Our study focuses on relation extraction tasks in the planetary science domain. Rovers and landers have

been exploring the surface of Mars for decades, and the science teams directing their activities have identified and named thousands of individual observation targets (rocks, soils, dunes, etc.). These targets are mentioned in subsequent scientific publications in conference and journal venues.

Our goal is to construct a relation extraction system that can successfully identify statements about the composition and properties of Mars targets. We assume that entities of type Target, Component (element or mineral), and Property (e.g., "layers", "dusty", "pits") have already been identified within the text, and the relation extraction system must determine which pairs of entities exhibit a given relation. We study two relations of interest: CONTAINS(*Target, Component*) and HASPROPERTY(*Target, Property*). An example of the CONTAINS relation was shown in Figure 1.

The sentence below includes three instances of the HASPROPERTY relation.

> *The dark rocks such as Barnacle Bill are more silica-rich, while the Bright Rocks such as Yogi and Wedge are more sulfur-rich and probably more weathered.*

The complete set of relations includes:

HASPROPERTY(*Barnacle Bill*, *dark*),

HASPROPERTY(*Yogi*, *weathered*),

HASPROPERTY(*Wedge*, *weathered*),

CONTAINS(*Barnacle Bill*, *silica*),

CONTAINS(*Yogi*, *sulfur*), and

CONTAINS(*Wedge*, *sulfur*).

It is common in this domain for multiple Targets to share a relation with the same Property (or the same Component in the CONTAINS relation). Conversely, it is also common for multiple Properties or Components to share a relation with a single Target. Relation extraction for this domain can be quite complex even when focusing on a single sentence. Additional challenges arise from the use of abbreviations for mineral names (e.g., Fe for iron), locally defined shorthand such as *BB* for *Barnacle Bill*, complex grammar with multiple clauses per sentence, and "hedging language" (Lakoff, 1972) that captures the uncertainty about properties of targets on another planet. Examples of hedging occur as the words "likely" and "possibly" in this sentence:
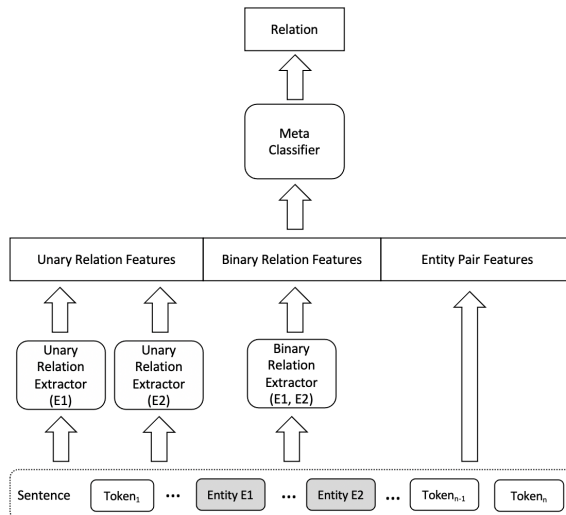

Figure 2: Stacked Learning Model

> *The Big Sky tailings were spectrally flat (similar to Telegraph Peak) likely from the presence of magnetite, and include a weak downturn > 750 nm, possibly from minor hematite.*

This complex sentence entails two relations:
CONTAINS(*Big Sky*, *magnetite*) and
CONTAINS(*Big Sky*, *hematite*).

4 Stacked Learning with Unary Relation Extractors

Our task is to perform within-sentence relation extraction given pre-specified ("gold") entities. We propose a stacked RE system in which a meta-classifier employs the output of a traditional binary relation extractor as well as two unary relation extractors, as shown in Figure 2. At a high level, the binary relation extractor captures the context spanning two arguments, while each unary relation extractor captures local information for one argument of the relation. The meta-classifier also includes features that describe the pair of entities under consideration. We utilize existing models from prior work for the binary relation extraction models. In this section, we present the design of our new unary relation extraction models and the meta-classifier.

4.1 The Stacked Learning Model

4.2 Unary Relation Extraction

A novel contribution of this work is the focus on *unary* relations. Each binary relation $\mathcal{R}(E_1, E_2)$ that applies to two entities can be decomposed into unary relations $\mathcal{R}_1(E_1)$ and $\mathcal{R}_2(E_2)$. Unary rela-

tions operate on a single entity to predict whether that entity acts as the appropriate argument for the relevant binary relation. For example, the sentence "*X provided guidance for Y's university studies*" includes an instance of the binary relation ADVISES(*Person, Person*) in the form of ADVISES*(X, Y)*. This relation can be decomposed into ADVISOR(*Person*) and ADVISEE(*Person*) which can be separately evaluated for each of *X* and *Y*. Local context may lead us to infer ADVISOR(*X*) and ADVISEE(*Y*), indicating their argument status and differentiating *X* and *Y* despite their identical entity types.

For the planetary science domain, we decompose CONTAINS(*Target, Component*) into two unary relations: (1) the CONTAINER unary relation focuses on Target entities that contain an unspecified component (i.e., CONTAINS(*Target, **)), and (2) the CONTAINEE unary relation focuses on Component entities that are part of an unspecified Target's composition (i.e., CONTAINS(**, Component*)).[2] Similarly, we decompose the HASPROPERTY(*Target, Property*) relation into (1) PROPERTYHOLDER, which corresponds to HASPROPERTY(*Target, **), and (2) PROPERTYHELD, which corresponds to HASPROPERTY(**, Property*). All of our unary relation extractors share the same model architecture, which is explained in the following section.

4.2.1 Unary Relation Extractors

Each unary relation extractor takes an entity along with its sentence as input and predicts whether the entity participates in a specific unary relation. We define the input sentence S to consist of n tokens $S_1, S_2, \ldots, S_n$ and represent the entity of interest, e, with its beginning and ending indices, *BGN(e)* and *END(e)* respectively. Following prior work (Zhong and Chen, 2021), we insert a begin marker $\langle B \rangle$ and an end marker $\langle /B \rangle$ around e in the sentence to highlight the entity of interest:

$$S' = ..., \langle B \rangle, S_{BGN(e)}, ..., S_{END(e)}, \langle /B \rangle, ...$$

We use a pre-trained language model to encode S' and produce a contextual representation for the sentence. We then use the representation of the start marker $\langle B \rangle$ as the entity representation, denoted as E. Intuitively, we expect the representation of the start marker to encode the relevant contextual evidence around the entity (e.g., "*an increase in*

---

[2]We use the asterisk (*) symbol to indicate when an argument is unspecified.

potassium"). We pass E into a ReLU activation layer, a dropout layer, and finally a single-layer neural network to produce a predicted probability for whether the entity participates in the unary relation.

4.2.2 Training the Unary Models

The positive training instances for unary relation $\mathcal{R}_i(T)$ consist of all annotated instances of entity type T that participate as argument i in a binary relation of type $\mathcal{R}$. Negative training instances consist of all instances of T that do not participate as argument i of a relation of type $\mathcal{R}$. We trained the extraction models by fine-tuning a pre-trained language model with cross-entropy loss: $L(\theta) = \sum_{\text{x, y} \in \text{train}} log P(y|x, \theta)$, where x is an instance, $y \in \{0, 1\}$ is the unary relation label, and θ are the model parameters. We experimented with several different language models, which we will discuss in Section 5.

We create one stacked model for each relevant pair of entity types (e.g., one model to extract relations for (*Target, Component*) and another for (*Target, Property*)). Each model takes a pair of entities (E_1, E_2) of types (T_1, T_2) in a sentence as input and produces a prediction for whether a relation of type $\mathcal{R}$ exists between E_1 and E_2 (see Figure 2). We represent each pair of entities with a feature vector based on three sets of features: unary relation features, binary relation features, and entity pair features.

4.2.3 Unary Relation Features

The unary relation features consist of the outputs (confidence scores) of both unary relation extractors and a "unary pairing" feature. The latter feature is true if either entity $E_i \in \{E_1, E_2\}$ satisfies the following two criteria: 1) E_i receives a confidence score of at least 50% for a unary relation $\mathcal{R}_i(T_i)$, and 2) E_i is the closest entity of type T_i relative to the other entity in the pair. Intuitively, this rule hypothesizes a probable binary relation when at least one of the entities is predicted to participate in a unary relation and no other entity of the same type is closer to the other entity.

More generally, if there are $k > 1$ relations for the same kind of entity pairs (e.g., ADVISES(*Person, Person*) or MARRIED(*Person, Person*)), the unary relation features consist of $2*k$ confidence scores and k unary pairing features.

4.2.4 Binary Relation Features

The binary relation feature is the output (confidence score) of the binary relation extractor. If there are $k > 1$ relations, a multi-class binary relation extractor is used to generate k posterior probabilities for the feature vector.

4.2.5 Entity Pair Features

The entity pair features capture general information about the context in which the entities occur.

Negation: Negation may suggest that there is no relation, so we create a binary feature to indicate if there is a negation word between E_1 and E_2.[3]

Order of Entities: One binary feature indicates the relative order of the two entities in the sentence.

Number of Entity Pairs: We count the number of entity pairs of type (T_1, T_2) in the sentence, and then bucket the counts into five bins $([1, 2), [2, 4), [4, 10), [10, \infty))$.

Nearest Entity: One binary feature indicates whether E_1 is the closest entity of type T_1 to E_2. Similarly, another binary feature indicates whether E_2 is the closest entity of type T_2 to E_1.

Entity Distribution: We hypothesize that the distribution of entities around E_1 and E_2 affects the likelihood that a relation exists between them. For example, a relation may be less likely if other entities occur between E_1 and E_2. So we develop two binary features to capture whether there is an entity of type T_2 to the left or to the right of E_1. Similarly, we develop two features to capture the same information for E_2 with type T_1.

Distance: Capturing the distance between two entities has shown to be useful in previous work. We create a distance feature by binning the number of words between the entities into q quantile bins, where the quantiles are computed over the distances observed in the training set. We explore different values $(2, 5, 10, 15, 20)$ for q (the number of bins), and choose the one that performs best on the development set.

4.2.6 Meta-classifier

The input to the meta-classifier is a sentence with two entities marked. We then create a feature vector based on the three aforementioned feature sets, and feed it into the meta-classifier to predict a relation. While the model choice is flexible, we used a linear SVM in our experiments.

---

[3]The negation words we use are: no, not, none, nothing, never, nowhere, hardly, barely, scarcely.

	Count
Documents	602
Targets	5,140
Components	15,826
Properties	14,895
CONTAINS	3,045
HASPROPERTY	2,764

Table 1: Annotation statistics for LPSC corpus.

5 Experiments

We conducted experiments to evaluate the stacked relation extraction approach on the planetary science document collection as well as a benchmark data set to compare directly with recent prior work. We release the dataset and codes at https://github.com/yyzhuang1991/StackedLearningWithUnaryModels.

5.1 Planetary Science (LPSC) Data Set

We used a total of 602 documents that were manually annotated by three planetary scientists from the Jet Propulsion Laboratory, who are also co-authors of this work, to indicate the presence of relevant entities (Target, Component, Property) and relationships between them as CONTAINS(Target, Component) or HASPROPERTY(Target, Property). The corpus consists of text extracted from publicly available two-page extended abstracts that were published at the Lunar and Planetary Science Conference (LPSC). We started with a public collection of 117 documents from 2015 and 2016 that were annotated with CONTAINS relations for targets from the Mars Science Laboratory mission (Francis and Wagstaff, 2017). To expand the collection size and have more than one relation to study, we annotated almost 500 additional documents from 1998 to 2020 for targets from three more Mars missions: Mars Pathfinder, Mars Phoenix Lander, and the MER-A (Spirit) Mars Exploration Rover (Wagstaff et al., 2022). We also added the new relation HASPROPERTY. Table 1 shows the total number of annotated entities and relations.

5.2 Planetary Science Domain Methodology

We randomly selected 25% of the documents (151 documents) for a development set to tune hyperparameters and performed 5-fold cross validation over the other 451 documents. We report the precision, recall, and F1 score averaged across the 5

runs. All of our relation extraction models take gold (manually annotated) entities as input.

To populate a knowledge base, it is sufficient to store each relation once. The planetary science domain experts who annotated the corpus had this mindset, so they did not always annotate duplicate instances of the same relation within a document. For example, if CONTAINS(*Windjana, potassium feldspar*) occurs three times in a document, they may have only annotated it once. Therefore, we report precision, recall, and F1 score by comparing the set of distinct predicted relations in a document against the set of distinct annotated relations, a metric that has been used in other RE work (Li et al., 2016b). Specifically, for each document and relation $\mathcal{R}$, we compiled the unique occurrences of entity pairs (E_1, E_2) annotated with relation $\mathcal{R}$ as the set of annotated relations. This approach ignores (and therefore does not require) duplicate occurrences of the same relation found in different sentences of a document.

5.3 Relation Extraction Models

We compared the performance of the stacked learning model to that of existing binary relation extractors as well as unary relation extraction alone. For binary relation extraction, we experimented with fine-tuning pre-trained language models (LMs) on our planetary science data, since previous work (Gu et al., 2022; Yasunaga et al., 2022) has found this approach to work quite well. We created models with BERT (Devlin et al., 2019), which has been widely used across many NLP tasks and domains, as well as SciBERT (Beltagy et al., 2019) and LinkBERT (Yasunaga et al., 2022), which have proven to be beneficial for other scientific domains.[4]

Specifically, each binary relation extraction system is a pre-trained language model with an additional classification layer on top. The input is a sentence with exactly two entities marked as relevant. The sentence is encoded by the language model, and its representation (as captured by the special [CLS] token) is then fed into a single layer feedforward network that produces a probability distribution over the set of possible relations.[5] For hyperparameters, we used a batch size of 10 and

a dropout rate of 0.1.[6] We then performed a grid search over all combinations of learning rates (1e-5, 2e-5) and epochs (4, 5, 10) and used the values which performed best on the development set.

To our knowledge, the only previous work on relation extraction for the planetary science domain was reported by Wagstaff et al. (2018), for the CONTAINS relation only. They used jSRE (Giuliano et al., 2006), which employs an SVM classifier to predict the presence of a relation between two entities given features derived from a shallow parser. For comparison with that earlier work in this domain, we also trained a jSRE model for each of our binary relations. For hyperparameter-tuning, we explored every possible combination of SVM kernels (LC, GC, SL) and window sizes (1, 2, 5, 10, 15, 20), choosing the values that performed best on the development set.

We also created a binary relation extraction system that uses *only* unary relation extractors. The challenge for this approach is how to recover binary relations from the unary predictions, particularly because multiple entities are often predicted for each unary relation. After exploring several strategies, the best approach seemed to be aggressively pairing each entity predicted to be in a unary relation with *all* other entities of the appropriate type in the sentence. For example, to produce the CONTAINS(Target, Component) relation, we pair each predicted CONTAINER(Target) with all Component entities, and likewise we pair each predicted CONTAINEE(Component) with all Target entities. We refer to this approach as the **Paired Unary (PU)** Model.

The logic behind this approach is that one unary relation can have strong local evidence, while the other may not. For example, this model performs well in situations where (say) two Targets are correctly recognized as CONTAINERS but the mineral detected at those sites is not recognized as being in a CONTAINEE context. We found that this heuristic worked fairly well in the planetary science domain, with substantially higher recall but lower precision. But an advantage of stacked learning is that it avoids the need to manually create heuristics because the meta-classifier learns what will work well in different domains and for different relations.

Model	Pr	Rec	F1
jSRE	**69.1**	75.8	72.3
PU$_{\text{SciBERT}}$	59.6	94.1	72.6
PU$_{\text{LinkBERT}}$	59.3	**95.1**	72.7
PU$_{\text{BERT}}$	60.2	94.6	73.2
BR$_{\text{SciBERT}}$	68.6	83.9	74.7
BR$_{\text{BERT}}$	66.5	87.9	75.0
BR$_{\text{LinkBERT}}$	68.4	86.6	**76.0**

Table 2: Results for the CONTAINS relation (LPSC)

.

Model	Pr	Rec	F1
jSRE	56.3	60.8	58.3
PU$_{\text{LinkBERT}}$	54.0	**97.1**	69.1
PU$_{\text{SciBERT}}$	55.4	95.8	69.8
PU$_{\text{BERT}}$	56.0	94.2	70.0
BR$_{\text{BERT}}$	74.5	75.0	74.6
BR$_{\text{SciBERT}}$	**76.0**	76.9	76.2
BR$_{\text{LinkBERT}}$	74.4	79.9	**76.9**

Table 3: Results for the HASPROPERTY relation (LPSC)

.

5.4 Results for Binary Relation Extraction

Table 2 shows the experimental results for jSRE, the Binary Relation (BR) extraction models, and the Paired Unary (PU) model for the CONTAINS relation. We fine-tuned the BERT, SciBERT, and LinkBERT language models for the planetary science domain for the PU and BR extractors.

The jSRE model achieved the highest precision but the lowest F1 score. All Paired Unary models slightly outperformed jSRE, but the Binary Relation models performed the best. Of the language models, BERT performed best for the PU models but LinkBERT performed best for the BR models.

Table 3 shows the results for the HASPROPERTY relation. The jSRE model performs substantially worse than for CONTAINS, and the performance of the PU models is lower too. However, the BR models achieve similar F1 scores, albeit with higher precision and lower recall than for CONTAINS. For both relations, LinkBERT is the best language model for binary relation extraction.

5.5 Results for Stacked Learning

For our stacked learning approach, we created a meta-classifier by training a linear SVM using the scikit-learn package (Pedregosa et al., 2011). We created three different stacked learning models con-

Model	Pr	Rec	F1
Stacked$_{\text{SciBERT}}$	67.5	86.9	75.6
Stacked$_{\text{BERT}}$	68.4	**89.3**	77.2
Stacked$_{\text{LinkBERT}}$	**72.1**	86.3	**78.5**

Table 4: Stacked learning results for CONTAINS.

Model	Pr	Rec	F1
Stacked$_{\text{BERT}}$	68.8	**86.2**	76.5
Stacked$_{\text{SciBERT}}$	71.2	85.0	77.4
Stacked$_{\text{LinkBERT}}$	**74.1**	82.8	**78.1**

Table 5: Stacked learning results for HASPROPERTY.

sisting of fine-tuned component models (unary and binary extractors) that all employed either BERT, SciBERT, or LinkBERT. For the SVM meta-classifier, we explored different values for the regularization parameter C within the set (0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5) and used the best value based on the development set.

Tables 4 and 5 show the results for stacked learning for the CONTAINS and HASPROPERTY relations respectively. The results are remarkably consistent. In every case, the stacked model that uses language model L performs better than the binary relation extractor that uses language model L. The best models use LinkBERT, where stacking improves performance from 76.0% → 78.5% for CONTAINS and from 76.9% → 78.1% for HASPROPERTY. These results demonstrate the value of combining unary and binary relation extractors in a stacked ensemble.

As a concrete example of the benefits of including unary relation extractors in the stacked model, consider the following sentence. It contains three CONTAINS relations between Target *Home Plate* and *Px* (an abbreviation for the mineral pyroxene), *Mt* (magnetite), and *npOx* (nanophase oxides). However, *Home Plate* does not contain *Ol* (olivine), which is a false positive for the BR$_{\text{LinkBERT}}$ model, but a true negative for Stacked$_{\text{LinkBERT}}$, which had access to the unary extractor and correctly predicted no CONTAINEE(*Ol*) relation.

> *Vesicular basalts investigated in the vicinity of Home Plate such as the rock Esperanza have the same Fe mineralogical composition as eastern Home Plate: rich in Px and Mt, no Ol and little npOx.*

5.6 Experiments on Chemprot Data

To assess the effectiveness of our approach in other domains, we conducted another experiment on the Chemprot task (Taboureau et al., 2010). The Chemprot task is designed to extract chemical-protein relations (CPR) from PubMed abstracts. In the dataset, five chemical-protein relations are used for evaluation (CPR:3, CPR:4, CPR:5, CPR:6 and CPR:9). We used code provided by Yasunaga et al. (2022) to obtain and preprocess the training, development, and test sets.[7] LinkBERT$_{\text{BioLinkBERT-base}}$ after fine-tuning is reported to achieve the best performance on this data set, so we used it as the pre-trained language model in unary extractors. The unary extractors and stacked model were trained as described in Section 5.3. A single stacked model is sufficient since all Chemprot relations operate on (*Chemical, Protein*) pairs. The binary relation features were generated by LinkBERT$_{\text{BioLinkBERT-base}}$ predictions kindly released by Yasunaga et al. (2022).

Table 6 shows the performance of different models, reported as precision, recall and F1 scores, micro-averaged and macro-averaged across the five relations. The Paired Unary model achieves the lowest overall performance due to its extremely low precision. It allows an entity to be mistakenly extracted by **multiple** unary relations in Chemprot. This emphasizes that heuristic rules to construct full relations using only the unary relation extractors may not work well for different domains.

The following rows show results reported by Gu et al. (2022) for competitive binary relation extractors produced by fine-tuning different language models.[8] Among these methods, LinkBERT achieves the best performance. The bottom row shows that our stacked model based on LinkBERT improves upon LinkBERT alone and achieves state-of-the-art performance on this task. Specifically, the micro-F1 score increases from 77.6% to 78.3% and the macro-F1 score from 76.8% to 77.9%. According to the precision-recall breakdown, the stacked model achieves a substantial increase in precision (by 3.2% absolute points in micro-average, and 5.7% in macro-average) although at the expense of some recall.

Model	Micro Average			Macro Average		
	P	R	F1	P	R	F1
PU$_{\text{LinkBERT}}$	36.3	**88.5**	51.5	34.5	**87.6**	49.4
Binary Extractors						
BERT[†]	-	-	71.8	-	-	-
SciBERT[†]	-	-	75.2	-	-	-
BioBERT[†]	-	-	76.1	-	-	-
PubMedBERT[†]	-	-	77.2	-	-	-
LinkBERT	76.6	78.5	77.6	75.6	76.1	76.8
Stacked Model						
SVM$_{\text{LinkBERT}}$	**79.8**	76.8	**78.3**	**81.3**	75.0	**77.9**

Table 6: Performance of relation extraction models on Chemprot. [†]: Scores reported by Gu et al. (2022).

Dataset	FULL	w/o UNARY	w/o EP	w/o BINARY
LPSC				
CONTAINS	78.5	77.5	77.2	75.3
HASPROPERTY	78.1	76.9	77.0	76.5
Chemprot				
CPR:3	75.6	74.5	75.6	65.5
CPR:4	81.8	81.7	81.8	78.6
CPR:5	81.3	78.2	81.6	67.3
CPR:6	82.6	82.9	81.3	75.1
CPR:9	68.2	67.6	68	62.9
Micro-AVG	78.3	77.6	78.2	72.7
Macro-AVG	77.9	77.0	77.6	70.0

Table 7: F1 scores of the stacked model SVM$_{\text{LinkBERT}}$ in ablation experiments.

6 Analysis

We performed additional manual analyses to better understand the behavior of our stacked models.

We performed ablation experiments to assess the contributions of different components of the stacked ensemble by separately removing the Unary Relation Features (UNARY), Binary Relation Features (BINARY), and Entity Pair Features (EP) from the best stacked model SVM$_{\text{LinkBERT}}$. Table 7 shows the F1 score for each relation within the LPSC and Chemprot data sets. For LPSC, removing any feature set reduces performance, so they are all valuable. For Chemprot, however, only BINARY and UNARY are important; excluding EP does not significantly impact the overall performance. Looking at individual CPR relations, we find that including unary relation features benefits CPR:3, CPR:5, and CPR:6 the most. This result suggests that those relations have more local contextual cues that are associated with one or the other side of the relation.

Next, we examined whether the stacked model extracts relations from sentences with more entities better. Figure 3 shows a graph that plots the F1 scores of SVM$_{\text{LinkBERT}}$ and LinkBERT against the number of entity pairs in a sentence for the
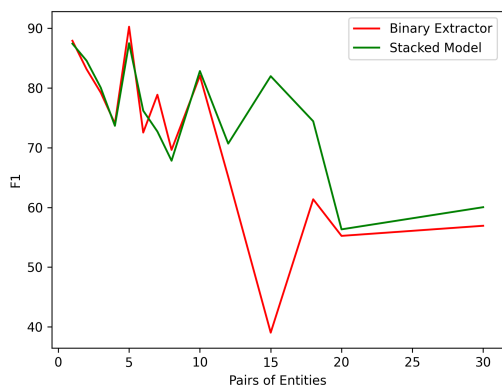
Figure 3: F1 scores versus the number of entity pairs (Target, Component) in a sentence for the CONTAINS relation. The binary relation extractor is LinkBERT and the stacked model is SVM$_{\text{LinkBERT}}$. Scores are averaged across the 5-fold cross validation.

CONTAINS relation. The stacked model performs comparably to the binary relation extractor over sentences with fewer entity pairs, but it consistently outperforms the binary relation extractor over sentences with more than 10 pairs of entities. We hypothesize that it is important to filter out entities that do not participate in any unary relation when there are many entities in a sentence. By recognizing unary relations, the stacked model is able to handle the complexity of a large number of entity pairs.

Finally, Table 8 shows some correct and incorrect cases extracted by SVM$_{\text{LinkBERT}}$. In the top portion, we show examples of the CONTAINS relation that the binary relation extractor, LinkBERT, missed but the stacked model correctly extracted. We found that a lot of these cases contain strong local cues (such as "*suggestive of*" in 1), and "*abundance*" in 2)) that signify relevant unary relations. The bottom portion of Table 8 shows some false positive examples where the stacked model incorrectly extracted the CONTAINS relation. 3) is a challenging case where the local context is misleading (e.g., "*Humphrey* contains cumulate *Olivine*") and it is important to understand the more global contexts *"there is not enough data"*. 4) is a common error we have observed, where both the Target and the Component entities are in the relevant unary relations but they do not participate in the same binary relation.

Missed ⟶ Extracted

1. An APXS analysis of the "*Hula*" sample (Figs.3, 4) shows elevated MgO (11wt%), SO3 (33wt%), and Ni (900 ppm), suggestive of Mg-*Nickel* sulfate.

2. The *Hematite* abundance (8 wt%) is significantly more than observed in other samples from Gale Crater: 0.8, 0.6, 0.7, and 0.6 wt% for Rocknest, John Klein , Cumberland , and *Windjana*, respectively[4, 5].

Falsely Extracted

3. At this time there is not enough data, experimental and petrologic, to suggest whether or not *Humphrey* contains cumulate *Olivine*.

4. Rock Humphrey shows similar *Phosphorus* contents in RU and RB and a decrease in RR , whereas for rock *Mazatzal* the highest P concentration is measured in its weathering rind of RB[5].

Table 8: Examples of correct and incorrect extraction by the stacked model, SVM$_{\text{LinkBERT}}$, for the CONTAINS relation. Components are highlighted in blue and Targets are highlighted in green.

7 Conclusions and Future Work

The goal of this work is to perform an automated analysis of scientific publications that enables the construction of domain-specific knowledge bases. We focused on the planetary science discipline, which to date has not received much attention from automated information extraction work. The complex grammar often employed in scientific publications can pose problems for state-of-the-art relation extraction systems. We proposed the use of unary relation extractors to enable specialization for each argument of a relation, within a stacked learning framework. Our approach performed well both in this domain and the Chemprot benchmark (biology) data set. In future work, we plan to expand the scope of this approach to include relations that cross sentences, which is a major challenge for current relation extraction systems and for which local unary relation modeling is especially well suited.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: Pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),*.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2018. A walk-based model on entity graphs for relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 81–88, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5308–5314, Florence, Italy. Association for Computational Linguistics.

Raymond Francis and Kiri L. Wagstaff. 2017. Mars Target Encyclopedia - LPSC abstracts labeled data set (Version 1.0.0 [Data set]). Presented at the Thirtieth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), https://doi.org/10.5281/zenodo.1048419.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pages 401–408.

Yuxian Gu, Robert Tinn, Hao Cheng, Michael R. Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3:1 – 23.

Sonal Gupta and Christopher Manning. 2011. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1–9, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 178–181, Barcelona, Spain. Association for Computational Linguistics.

George Lakoff. 1972. *The Semantics of Natural Language*, chapter Linguistics and natural logic. D. Reidel Publishing Company, Dordrecht, Holland.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, 18.

Jiao Li, Yueping Sun, Robin Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn Mattingly, Thomas Wiegers, and Zhiyong lu. 2016a. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016b. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016. Baw068.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. 2015. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 635–644, Beijing, China. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea. Association for Computational Linguistics.

Olivier Taboureau, Sonny Kim Nielsen, Karine Audouze, Nils Weinhold, Daniel Edsgärd, Francisco S Roque, Irene Kouskoumvekaki, Alina Bora, Ramona Curpan, Thomas Skøt Jensen, et al. 2010. Chemprot: a disease chemical biology database. *Nucleic Acids Research*, 39(suppl_1):D367–D372.

Yuka Tateisi, Yo Shidahara, Yusuke Miyao, and Akiko Aizawa. 2014. Annotation of computer science papers for semantic relation extrac-tion. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1423–1429, Reykjavik, Iceland. European Language Resources Association (ELRA).

Chen-Tse Tsai, Gourab Kundu, and Dan Roth. 2013. Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, page 1733–1738, New York, NY, USA. Association for Computing Machinery.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884, New Orleans, Louisiana. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Kiri L. Wagstaff, Raymond Francis, Matthew Golombek, Leslie Tamppari, and Steven Lu. 2022. Mars Target Encyclopedia - Labeled LPSC abstracts for four Mars missions (Version 1.0.0 [Data set]). https://doi.org/10.5281/zenodo.7066107.

Kiri L. Wagstaff, Raymond Francis, Thamme Gowda, You Lu, Ellen Riloff, Karanjeet Singh, and Nina L. Lanza. 2018. Mars Target Encyclopedia: Rock and soil composition extracted from the literature. In *Proceedings of the Thirtieth Annual Conference on Innovative Applications of Artificial Intelligence*, pages 7861–7866.

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 Annual Meeting of the Association for Computational Linguistics*, pages 1371–1377, Florence, Italy. Association for Computational Linguistics.

Taiki Watanabe, Akihiro Tamura, Takashi Ninomiya, Takuya Makino, and Tomoya Iwakura. 2019. Multi-task learning for chemical named entity recognition with chemical compound paraphrasing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6244–6249, Hong Kong, China. Association for Computational Linguistics.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. In *Association for Computational Linguistics (ACL)*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Joural of Machine Learning Research*, 3:1083–1106.

Ce Zhang, Vidhya Govindaraju, Jackson Borchardt, Tim Foltz, Christopher Ré, and Shanan Peters. 2013. GeoDeepDive: Statistical inference using familiar data-processing languages. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 993–996.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. Preprint on arXiv, https://arxiv.org/abs/1508.01006v2.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.