

An Instance Level Approach for Shallow Semantic Parsing in Scientific Procedural Text

Daivik Swarup, Ahsaas Bajaj, Sheshera Mysore
Tim O’Gorman, Rajarshi Das, Andrew McCallum

{dswarupogguv, abajaj, smysore
togorman, rajarshi, mccallum}@cs.umass.edu

Abstract

In specific domains, such as procedural scientific text, human labeled data for shallow semantic parsing is especially limited and expensive to create. Fortunately, such specific domains often use rather formulaic writing, such that the different ways of expressing relations in a small number of grammatically similar labeled sentences may provide high coverage of semantic structures in the corpus, through an appropriately rich similarity metric. In light of this opportunity, this paper explores an instance-based approach to the relation prediction sub-task within shallow semantic parsing, in which semantic labels from structurally similar sentences in the training set are copied to test sentences. Candidate similar sentences are retrieved using SciBERT embeddings. For labels where it is possible to copy from a similar sentence we employ an instance level copy network, when this is not possible, a globally shared parametric model is employed. Experiments show our approach outperforms both baseline and prior methods by 0.75 to 3 F1 absolute in the Wet Lab Protocol Corpus and 1 F1 absolute in the Materials Science Procedural Text Corpus.

1 Introduction

Being able to represent natural language descriptions of scientific experiments in a structured form promises to allow tackling a range of challenges from automating biomedical experimental protocols (Kulkarni et al., 2018) to gaining materials science insight by large scale mining of the literature (Mysore et al., 2019). To facilitate these applications, recent work has created datasets annotated with sentence level semantic structure for procedural scientific text from experimental biology (Kulkarni et al., 2018) and materials science (Mysore et al., 2019). However, these corpora, the Wet Lab Protocols corpus (WLP) and the Materials

Query: “Centrifuge the sample at 14,000xg for 5 minutes.”
Neighbor: “Centrifuge supernatant at 12,000xg for 10 minutes.”
Query: “Add 700 μ l 70% ethanol to the tube and invert several times to wash the DNA pellet.”
Neighbor: “Add 200 μ l 70% ethanol and invert the tube twice to wash the pellet.”

Figure 1: Example sentences from the WLP corpus, and their nearest neighbours based on sentence representations obtained from SciBERT.

Science Procedural Text (MSPT) corpus remain small. This motivates approaches to parsing that are likely to generalize given limited labelled data.

We propose an instance-based edge-factored approach for the relation prediction sub-problem of shallow semantic parsing. To predict a possible relation between two entities, our approach retrieves a set of sentences similar to the target sentence, and learns to copy relations in those sentences to the target sentence (Figure 1 shows some examples).

However, using only a nearest-neighbours approach over similar sentences poses a coverage problem, as some edge labels may have zero instances in the set of nearest neighbour sentences. To address this, we employ a parametric approach which can score a label when it is not possible to copy that label from any of the neighbours. Therefore, we combine a local, instance-level approach with a global, parametric approach.

Our instance-based approach is motivated by the observation that text in the WLP and MSPT corpora, both of which describe experimental protocols, follow domain-specific writing conventions (sometimes referred to as a *sublanguage* (Grish-

man, 2001; Grishman and Kittredge, 1986)) resulting in text that is repetitive and semi-structured. In such restricted domains we postulate that a low-bias instance-level approach may generalize better compared to a parametric approach, which is likely to suffer from a lack of training data.

In evaluations of the proposed approach we find the proposed local and global approach to outperform baseline methods based on parametric approaches by 0.75 F1 absolute in WLP and 1 F1 absolute in MSPT and prior work by 2.69 F1 absolute (12.7 % error reduction) on the WLP corpus. We also present first results for relation prediction on the MSPT corpus. Code and data for our experiments is available.¹

2 Task Setup and Notation

Given a sentence $X = \langle x_1, \dots, x_i, \dots, x_L \rangle$ from a dataset $\mathcal{D}$, let x denote tokens, and (m, t) entity mentions and their entity types, where $m \in C$, where C is the set of all possible contiguous token spans in X .² In a sentence, we denote the set of all entity mentions with M . Given this, we focus on the task of *relation prediction* which outputs a set of directed edges E such that, $e = (m_s, m_d, r)$ with $e \in E \subset M \times M$, where m_s, m_d denote source and destination mentions, $r \in \{\mathcal{R} \cup \emptyset\}$ denotes a relation edge label, $\mathcal{R}$ denotes the set of relation labels defined for the dataset and $\emptyset$ denotes the absence of a relation.

3 Local and Global Model for Relation Prediction

The proposed relation prediction approach is a combination of two components: a local, instance-based component which predicts the relation r of one edge (m_s, m_d) by copying a label from a set of nearest neighbor edges $e_n = (m_{ns}, m_{nd}, r_n) \in N$, and a second component making a prediction from a globally shared set of parameters. The set of nearest neighbor edges N is obtained from similar sentences in the training set (§3.2). This is formulated as follows:

$$P_{lg}(r_i | m_s, m_d, N) = \begin{cases} \frac{1}{Z} e^{E_l(r_i, m_s, m_d, N)} & \text{if } r_i \in \text{labels}(N) \\ \frac{1}{Z} e^{E_g(r_i, m_s, m_d)} & \text{if } r_i \notin \text{labels}(N) \end{cases} \quad (1)$$

¹<https://github.com/bajajahsaas/knn-srl-procedural-text>

²Non-contiguous entities in WLP (< 1%) are excluded.

Here, E_g represents the globally shared scoring function and E_l the local scoring function, here we drop additional arguments to these functions for brevity. Z denotes the normalization constant where: $Z = \sum_{r_k \in \text{labels}(N)} e^{E_l(r_k)} + \sum_{r_j \notin \text{labels}(N)} e^{E_g(r_j)}$. In computing the score from E_l per label, an instance level score from $E_c(r_i, m_s, m_d, e_n)$ is aggregated for every label present in the neighbours N as: $E_l = \text{logsumexp}_{\text{label}(e_n)=r_i} E_c$. This represents making a soft maximum selection of a neighbour edge most similar to the test edge for a given label r_i . Here, $\text{labels}(N)$ returns the set of labels present in N and $\text{label}(e_n)$, returns the neighbour edge label.

Equation 1 represents a model which is biased first to copy edge labels from N and in the absence of a label in N rely on a global model. This is in contrast to a model which trades off local and global models in a data dependent manner, the approach taken in the copy-generate model of See et al. (2017). The proposed formulation imposes an inductive bias in the model to copy edge labels which we believe helps perform well in our small data regime. In practice, our approach uses the local model for more frequently occurring labels and the global model for rare labels. Conceptually, this is once again, in contrast to the models of See et al. (2017) and Gu et al. (2016) which use a copy-model for long-tail or low-frequency phenomena. We believe this contrast is reasonable due to the formulaic nature of the text and the small data regime. Here, a local instance-level approach is able to generalize better by copying labels while the global model suffers from a lack of training data to learn the majority label patterns. Low frequency labels would see comparable performance for the global and instance level models. We confirm these intuitions empirically in §4. Next we define the neural-network parameterization of the model.

3.1 Edge Representation and Scoring Function Parameterization

We define the instance level scoring function E_c and E_g for the global model as follows:

$$E_c(e_n) = \text{FFN}_R([e_q; e_n; r_n]) \quad (2a)$$

$$e_q = \text{FFN}_e([m_s; m_d; t_s; t_d; d_{s,d}]) \quad (2b)$$

$$e_n = \text{FFN}_e([m_{ns}; m_{nd}; t_{ns}; t_{nd}; d_{ns,nd}]) \quad (2c)$$

Here, FFN_R is a feed-forward network which returns a scalar, e_q the vector representations for the

query/test edge, $\mathbf{e}_n$ the neighbour edge and $\mathbf{r}_n$ the neighbours relation. Network FFN_e produces a vector representations for e_q or e_n . And, $\mathbf{m}$ represents a contextualized representation for the source and destination entity mentions, $\mathbf{t}$ and $\mathbf{d}$ represents a vector representations of the entity type and the distance between the source and destination. The parameters $\mathbf{t}$, $\mathbf{r}$ and $\mathbf{d}$ are learned as model parameters and contextualized mention representations are obtained from SciBERT (Beltagy et al., 2019) (word-pieces averaged) without fine-tuning. Next, the global scoring function is formulated as:

$$E_g(r_i) = \text{FFN}_R([\mathbf{e}_q; \mathbf{e}_{r_i}; \mathbf{r}_i]) \quad (3)$$

While most notation remains the same as in Equation 2, $\mathbf{e}_{r_i}$ represents a globally shared “prototype” edge representation per label, learned as model parameters. Note that $\mathbf{e}_{r_i}$ is only used in the global model and is the same kind of object as $\mathbf{e}_n$.

3.2 Training and Sentence Retrieval

The proposed approach is trained by maximizing the log likelihood of the observed relations, r^* in the dataset: $\mathcal{L} = \sum_{\mathcal{D}} \sum_E \log \text{Pl}_g(r^*)$

In this work, we obtain the set of nearest neighbour sentences to obtain N based on representations obtained from SciBERT. Every sentence is represented by the average of the token (word-piece) representations: $\mathbf{v}_X = \frac{1}{L} \sum_{i=1}^L \text{SciBERT}(x_i)$. K nearest neighbours of the query sentence X_q were ranked by scores obtained as: $\text{cosine_sim}(\mathbf{v}_{X_q}, \mathbf{v}_{X_n})$. We set $K = 5$ at training time to obtain the set of edges, N . At test time we use $K = 40$ and $K = 20$ for WLP and MSPT respectively. In experiments, we work with approximate nearest neighbours obtained from the annoy package.³ Complete model hyperparameter and training details are presented in Appendix A.4.

4 Results and Analysis

We evaluate the proposed approach on two datasets of procedural scientific text: the Materials Science Procedural Text (MSPT) corpus and the Wet Lab Protocols (WLP) corpus. In both corpora we focus on the sentence level relation prediction task given gold entity mention spans. The experimental setup is detailed in Appendix A.1.

³<https://github.com/spotify/annoy>

4.1 Baselines

We compare the proposed approach to several baseline approaches as well as prior work:

KULKARNI18: The best approach proposed in prior work on the WLP corpus. This is an edge factored parametric approach using lexical, dependency and entity-type features.

COPYGEN: This is the copy-generate model proposed in (See et al., 2017), modified for a relation prediction task. The method differs from ours in trying to predict a copy probability, α using a mixing network which trades off the copy/instance or generate/global component in a data-dependent manner. The model is detailed in Appendix A.2.1.

STRINGCOPY: This approach attempts to copy the relation for a query edge (m_{qs}, m_{qd}) from a neighbour edge (m_{ns}, m_{nd}), from the nearest neighbours N , first based on exact string matches of the mention and next the entity type t . If this is not possible it predicts $\emptyset$.

GLOBALMODEL: A parametric model approach without an instance learning component: $P_g(r|m_s, m_d) = \text{Softmax}(\text{FFN}_g(\mathbf{e}_q))$. Since this is the dominant approach for relation prediction we believe it is the most reasonable relation prediction model to compare against to demonstrate the benefits of an instance learning approach.

LOCALMODEL: Instance based local approach (Eq 1) without the global model.

4.2 Results

Overall results: Table 1 presents performance of the proposed approach against a host of baseline methods and prior work. From row I, we note that the inductive bias to copy is better suited to WLP than to MSPT, and that simple rule-based approaches don’t perform at any useful level. Also note the proposed approach outperforms prior work on WLP (II vs VI). Next, we note that the parametric and the instance based approach (IV, V) trade off precision and recall as we would expect and that the proposed approach (VI) outperforms both these approaches. Also note the ablation of model components provided in this result (IV, V, VI).

Next consider specifically the results on MSPT. Note here, the high-recall result of COPYGEN. We explain this as follows: First we note that given the formulaic nature of the data, the proposed approach is biased to have a higher precision given that it can copy labels. The COPYGEN and GLOBALMODELS lack this bias. The MSPT dataset has a sparser set

ID	Model	WLP			MSPT		
		Precision	Recall	F1	Precision	Recall	F1
I	STRINGCOPY	6.99	35.45	11.68	1.42	15.71	2.61
II	KULKARNI18	80.98	77.04	78.96	-	-	-
III	COPYGEN	81.17	80.59	80.88	66.33	72.14	69.11
IV	GLOBALMODEL	81.06	80.77	80.91	66.93	70.66	68.75
V	LOCALMODEL	81.32	78.75	80.01	68.72	64.16	66.36
VI	OUR METHOD	82.29	81.02	81.65	70.04	69.48	69.76

Table 1: Our methods compared against baseline approaches and prior work on the test sets of the Web Labs Protocols (WLP) and Material Science Procedural Text (MSPT) corpora. Results assume access to gold entity mentions and represent microaveraged performance.

of relations when considering all pairs of edges between entity mentions ($1916/45732 = 4.1\%$) than WLP ($8264/60338 = 13.6\%$). To perform well on a sparsely labelled dataset a model must be biased for precision (a conservative model biased for precision would label the true-positives and given the sparsity, have high recall and overall F1), since the COPYGEN/GLOBALMODELS models are not biased for precision they make predictions more liberally leading to higher recalls but see significant hits to precision, in contrast to the proposed method. Finally, we note the gap between CopyGen and GlobalModel in MSPT and attribute it to training variance given the smaller size of MSPT.

Finally, we also compare to an alternative data-dependent method for combining a parametric and instance based approach (III vs VI) from See et al. (2017). Our approach with a stronger inductive bias to copy relations outperforms this. We also note that this approach performs similarly to GLOBALMODEL (III vs IV). Examination of the predicted copy-probability (α) on development examples in COPYGEN shows these values to be very small (MSPT mean: 10^{-5} , WLP mean: 10^{-5}) confirming that the model always chooses to “generate” (i.e. use a parametric model) and lacks sufficient inductive bias to copy in our datasets. In contrast, in OUR METHOD the local model makes edge predictions in 1852 of 1916 edges (96%) in MSPT and 8131 of 8264 edges (98%) in WLP development sets. Confirming the intended and significant invocation of the local model in the proposed approach.

Breakdown by label: As discussed in §3, given our small data regime, we believe a model with a simple inductive bias such as the local model generalizes better while the global model suffers a lack of training data to learn the majority label patterns, while in the case of very low frequency

		Data %	5	10	20	50	100
WLP	GM		69.18	72.72	76.78	78.76	80.91
	OM		70.32	73.64	77.12	79.24	81.65
MSPT	GM		48.87	57.96	61.88	65.83	68.75
	OM		50.8	59.17	60.82	66.42	69.76

Table 2: Performance of GLOBALMODEL (GM) compared against the OUR METHOD (OM) with varying amounts of training data on test F1.

labels the global component would perform at par with a simple parametric approach. We see this behaviour in Table 3. While this behaviour reverses the trend of methodologically similar instance based approaches (See et al., 2017; Snell et al., 2017; Khandelwal et al., 2020), we believe it to be reasonable specifically due to the formulaic writing in our corpora.

Varying training data: Finally, in Table 2 we note that the the proposed approach outperforms the parametric approach, GLOBALMODEL, at nearly all levels of training data. Demonstrating that the gains from copying labels from similar sentences in the training data hold out even as the pool of sentences to copy from shrinks, once again demonstrating the advantage of a model leveraging formulaic writing.

5 Related Work

Instance-based learning approaches have been applied to a range number of information extraction tasks such as Semantic Role Labeling (SRL), Named Entity Recognition (NER), and Part of Speech (POS) tagging. Akbik and Li (2016) and Wiseman and Stratos (2019) presents closest related work in terms of the task instance level methods are applied to. Akbik and Li (2016) apply a nearest-neighbors model for the SRL tasks of predicate and argument labeling based on pre-defined

WLP	Acts-on	Using	Mod-Link	Meronym	Creates	Count
Count	2589	1015	708	345	93	80
OUR METHOD	86.51	72.34	88.72	53.66	23.44	82.76
GLOBALMODEL	85.71	70.75	87.84	58.06	35.48	81.38
MSPT	Participant	Amount	Precursor	Condition	Target	Type
Count	395	375	196	135	84	33
OUR METHOD	64.54	78.42	54.16	29.91	51.46	76.67
GLOBALMODEL	61.09	77.53	58.99	20.1	50.85	80

Table 3: Per label performance for OUR METHOD compared against the GLOBAL MODEL on a random subset of labels in each dataset sorted by test set count/frequency. Total label instances, WLP: 8563, MSPT: 3119

feature representations of predicate-argument pairs; our work presents an instance level approach for the argument-labeling sub-task. Wiseman and Stratos (2019) applied instance-based methods to the sequence labeling tasks of NER and POS tagging, copying nearest neighbor labels from a set of candidate sentences as in the current work but applied to text spans. More generally, instance-based methods have also proven useful for language modeling (Khandelwal et al., 2020), knowledge base reasoning tasks (Das et al., 2020), and few-shot classification (Snell et al., 2017; Sung et al., 2018) and regression (Quinlan, 1993) problems.

Works in text generation such as summarization (See et al., 2017; Gu et al., 2016) have also incorporated “copy” mechanisms, pointing at long-tail phenomena from text to be summarized or translated rather than directly predicting them. These methods bear close methodological similarity to the proposed approach while differing in having a weaker inductive bias to copy labels. Also similar, are retrieve-and-edit approaches which have been applied instance based methods for generating complex structured outputs and text generation (Hashimoto et al., 2018; Guu et al., 2018).

6 Conclusion

We propose an edge factored instance based approach to the relation prediction sub-task within shallow semantic parsing for procedural scientific text. Our approach leverages the highly formulaic writing of procedural scientific text to achieve better generalization than baseline methods with weaker inductive biases to copy and prior approaches which represent parametric approaches on two corpora of English scientific text. While our work has only looked at predicting relations in an edge factored manner future work might explore ways of predicting higher order groups of edges.

Other extensions might consider jointly predicting spans and edges as in Akbik and Li (2016). Future work might also consider questions of characterizing and measuring formulaicity in text and how a range of information extraction tasks may be tailored to these texts. Finally, our approach relies on a static retrieval of sentences, there may also be potential for this aspect to be improved upon with a dynamic retrieval model trained along side the label prediction models similar to Guu et al. (2020), we expect this would be feasible particularly given the small dataset sizes in this domain.

Acknowledgments

We thank anonymous reviewers and members of UMass IESL group for helpful discussion and feedback. This work is funded in part by the Center for Data Science and the Center for Intelligent Information Retrieval, in part by the National Science Foundation under Grants No. IIS-1763618 and DMR-1922090, in part by USC (University of Southern California) subcontract no. 123875727 under Office of Naval Research (ONR) prime contract no. N660011924032, and in part by the Chan Zuckerberg Initiative under the project Scientific Knowledge Base Construction. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

- Alan Akbik and Yunyao Li. 2016. [K-SRL: Instance-based learning for semantic role labeling](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [Scibert: Pretrained language model for scientific text](#). In *EMNLP*.

- Rajarshi Das, Ameya Godbole, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2020. [A simple approach to case-based reasoning in knowledge bases](#). In *Automated Knowledge Base Construction*.
- Ralph Grishman. 2001. Adaptive information extraction and sublanguage analysis. In *Proceedings of the Workshop on Adaptive Text Extraction and Mining, Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, Washington, August 5, 2001*.
- Ralph Grishman and Richard Kittredge. 1986. *Analyzing language in restricted domains: sublanguage description and processing*. Psychology Press.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. [Incorporating copying mechanism in sequence-to-sequence learning](#). In *ACL*.
- Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. [Generating sentences by editing prototypes](#). *Transactions of the Association for Computational Linguistics*.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Papisut, and Ming-Wei Chang. 2020. [REALM: Retrieval-augmented language model pre-training](#). In *Proceedings of the International Conference on Machine Learning 1 pre-proceedings (ICML)*.
- Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. [A Retrieve-and-Edit framework for predicting structured outputs](#). In *Advances in Neural Information Processing Systems*, pages 10052–10062.
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. [Generalization through memorization: Nearest neighbor language models](#). In *International Conference on Learning Representations*.
- Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. 2018. [An annotated corpus for machine reading of instructions in wet lab protocols](#). In *Proceedings of NAACL-HLT*.
- Sheshera Mysore, Zach 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*. Association for Computational Linguistics.
- J Ross Quinlan. 1993. Combining instance-based and model-based learning. In *Proceedings of the tenth international conference on machine learning*, pages 236–243.
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Jake Snell, Kevin Swersky, and Richard Zemel. 2017. [Prototypical networks for few-shot learning](#). In *Advances in neural information processing systems*.
- Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. [Learning to compare: Relation network for few-shot learning](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Sam Wiseman and Karl Stratos. 2019. [Label-agnostic sequence labeling by copying nearest neighbors](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

A Appendix

A.1 Experimental Setup

WLP: We perform experiments with the splits provided by Kulkarni et al. (2018). In processing the dataset, we also exclude the “Misc-Link” as recommended, and cross sentence relations and relations with non-contiguous entities ($< 0.1\%$).

MSPT: We use data and `sfix` splits provided as the alongside Mysore et al. (2019).⁴ A small number of relations labelled across sentences ($< 1\%$) were removed.

A.2 Baseline Descriptions

A.2.1 Copy-Generate Based Relation Prediction

The COPYGEN forms one of our baseline approaches and bears similarity to the pointer-generator network proposed by See et al. (2017) for text summarization.

Here one component attempts to predict edges given entity mentions $m_s, m_d \in M$ and another which attempts to copy an edge relation label for (m_s, m_d) from a set of edges, $e_n = (m_{ns}, m_{nd}, r_n) \in N$ obtained from nearest neighbour sentences to the current sentence from the training set. This model is formulated as follows:

$$\begin{aligned} P_{cg}(r_i|m_s, m_d, N) &= \alpha P_{copy}(r_i|m_s, m_d, N) \\ &\quad + (1 - \alpha) P_{gen}(r_i|m_s, m_d) \\ \alpha &= \sigma(E_m(m_s, m_d, N)) \end{aligned}$$

Here, $\alpha \in [0, 1]$ denotes a mixing factor for the copy and generate models, σ denotes the sigmoid function, E_m denotes the mixing network and P_{cg} , P_{copy} and P_{gen} denote the copy-generate, copy and generate models respectively. These individual models are defined as follows:

$$\begin{aligned} P_{gen}(r_i|m_s, m_d) &= \frac{e^{E_g(r_i, m_s, m_d)}}{\sum_{j=1}^{|\mathcal{R}|+1} e^{E_g(r_j, m_s, m_d)}} \\ P_{copy}(r_i|m_s, m_d, N) &= \sum_{r_{nk}=r_i} P_{att}(a_k|m_s, m_d, N) \\ P_{att}(a_k|m_s, m_d, N) &= \frac{e^{E_c(a_k, m_s, m_d, N)}}{\sum_{k=1}^{|N|} e^{E_c(a_k, m_s, m_d, N)}} \end{aligned}$$

Here, E_g and E_c denote the generate and copy scoring functions respectively, and P_{att} denotes an attention distribution over edges (N) from the nearest neighbour sentences. While E_g and E_c are

formulated similar to those in Section 3.1, E_m is formulated as follows:

$$\begin{aligned} \alpha &= \text{FFN}_m([e_g; \mathbf{N}]) \\ \mathbf{N} &= \sum_{k=1}^{|N|} P_{att}(a_k) \mathbf{e}_{nk} \end{aligned}$$

Here, FFN_m yields scalar mixing scores based on the current edge representation e_g and a representation of the nearest neighbor set $\mathbf{N}$ obtained as a attention weighted sum of the neighbor edge representations.

A.3 Extended Results

While Table 1 presented test set results we include performance on the development set in Table 4.

A.4 Hyperparameters and Compute Details

Table 5 shows the choice of hyper parameters. We did not tune any hyperparameters other than the number of nearest neighbors. We evaluated the models for the following values of K : $\{5, 10, 15, 20, 30, 40, 50\}$ and chose the K with the best validation set F1 score for each dataset. During training, we only use $K = 5$. We ran experiments on server nodes with 256G RAM on a single Nvidia TITAN X GPU. Training models on the MSPT and WLP corpora took about 3 and 3-5 hours respectively.

⁴<https://github.com/olivettigroup/annotated-materials-syntheses>

	WLP			MSPT		
	Precision	Recall	F1	Precision	Recall	F1
STRINGCOPY	5.81	31.23	9.80	1.31	14.77	2.40
COPYGEN	80.83	79.95	80.39	66.45	72.49	69.34
GLOBALMODEL	80.75	79.06	79.9	67.22	69.95	68.56
LOCALMODEL	80.76	76.12	78.38	67.24	63.15	65.13
OUR METHOD	81.06	80.77	80.91	70.3	68.02	69.14

Table 4: Our methods compared against baseline approaches and prior work on the validation sets of the Web Labs Protocols (WLP) and Material Science Procedural Text (MSPT) corpora. Results assume access to gold entity mentions and represent microaveraged performance.

Parameter	WLP	MSPT
FFN_R^{**}	$768 \times 512 \times 256 \times 1$	$512 \times 256 \times 128 \times 1$
FFN_e^{**}	$1920 \times 512 \times 256 \times 256$	$1920 \times 256 \times 128 \times 128$
FFN_m^{**}	$256 \times 256 \times 126 \times 64 \times 1$	$128 \times 256 \times 126 \times 64 \times 1$
FFN_g^{**}	$256 \times 512 \times 256 \times 14$	$128 \times 256 \times 128 \times 19$
Distance Feature Buckets*	11	10
Number of Neighbors (Training)	5	5
Number of Neighbors (Testing)	40	20
Distance Feature Size (d)	128	128
Type Embedding Size (t)	128	128
Relation Embedding Size (r)	256	256
Learning rate	1×10^{-4}	1×10^{-4}
Weight decay	1×10^{-4}	1×10^{-4}
Optimizer	ADAM	ADAM

Table 5: Hyperparameter settings for models. *Number of tokens between source and destination entities are bucketed. We take the range of distances up to the 90th percentile and divide it into equal buckets. Instances with greater distance than this range fall into the largest bucket. ** All feed forward networks use ReLU non-linearities between layers.