AIFB-WebScience at SemEval-2022 Task 12: Relation Extraction First - Using Relation Extraction to Identify Entities

Nicholas Popovic
Karlsruhe Institute of Technology (KIT), Germany
popovic@kit.edu

Walter Laurito
FZI Research Center for Information Technology, Germany
laurito@fzi.de

Michael Färber
Karlsruhe Institute of Technology (KIT), Germany
michael.faerber@kit.edu

Abstract

In this paper, we present an end-to-end joint entity and relation extraction approach based on transformer-based language models. We apply the model to the task of linking mathematical symbols to their descriptions in LaTeX documents. In contrast to existing approaches, which perform entity and relation extraction in sequence, our system incorporates information from relation extraction into entity extraction. This means that the system can be trained even on data sets where only a subset of all valid entity spans is annotated. We provide an extensive evaluation of the proposed system and its strengths and weaknesses. Our approach, which can be scaled dynamically in computational complexity at inference time, produces predictions with high precision and reaches 3rd place in the leaderboard of SemEval-2022 Task 12. For inputs in the domain of physics and math, it achieves high relation extraction macro F_1 scores of 95.43% and 79.17%, respectively. The code used for training and evaluating our models is available on GitHub[1].

1 Introduction

Information extraction systems are a key component in making scientific literature more consumable. With the large amount of scientific works which are constantly being published (e.g., more than 60,000 machine learning papers per year (Färber, 2019)), indexing techniques that go beyond keyword searches are becoming more important. While many efforts have focused on the processing of abstracts as a way of building representations of publications (Gábor et al., 2018; Luan et al., 2018), methods processing full text documents will be needed to accurately capture their contents for use cases such as academic search and recommender systems and scientific impact quantification.

The task tackled in this paper (Lai et al., 2022), consisting of linking mathematical symbols to their descriptions in LaTeX documents, is a *joint entity and relation extraction* task. While earlier work tackled both subtasks sequentially via separate models, more recent approaches tend to use a single joint model (Luan et al., 2018; Bekoulis et al., 2018; Nguyen and Verspoor, 2019; Eberts and Ulges, 2021). In contrast to early approaches, which are based on Bi-LSTMs (Luan et al., 2018; Bekoulis et al., 2018; Nguyen and Verspoor, 2019), more recent approaches (Wadden et al., 2019; Eberts and Ulges, 2021) make use of transformer-based language models, such as BERT (Devlin et al., 2019). A key challenge in joint models is the computational complexity stemming from pairwise comparisons between entity spans required for relation extraction. Previous works tackle this using a span scoring mechanism based on a feed forward neural network, which produces a score indicating the likelihood that a span is in a relation (Luan et al., 2018; Wadden et al., 2019). Relation extraction is then performed on only those spans with the highest scores. For data sets which include span annotations even for entities which are not in any relation, such as DocRED (Yao et al., 2019), as examined by Eberts and Ulges (2021), such a scoring mechanism is not necessary, because the entity extraction component of the model can be trained on these annotations. For the task tackled in this paper, complete annotations for entity spans are not provided, making the use of a span scoring mechanism necessary.

In this paper, we propose an end-to-end approach for joint entity and relation extraction. The approach is based on a transformer-based language model, following previous work (Eberts and Ulges, 2020, 2021), but is peculiar in the sense that it incorporates a span scoring mechanism based on dot product similarity which is learned via triplet loss rather than cross entropy loss, making it applicable to datasets which contain annotations only for a subset of all valid entity mention spans.

[1] https://github.com/nicpopovic/RE1st

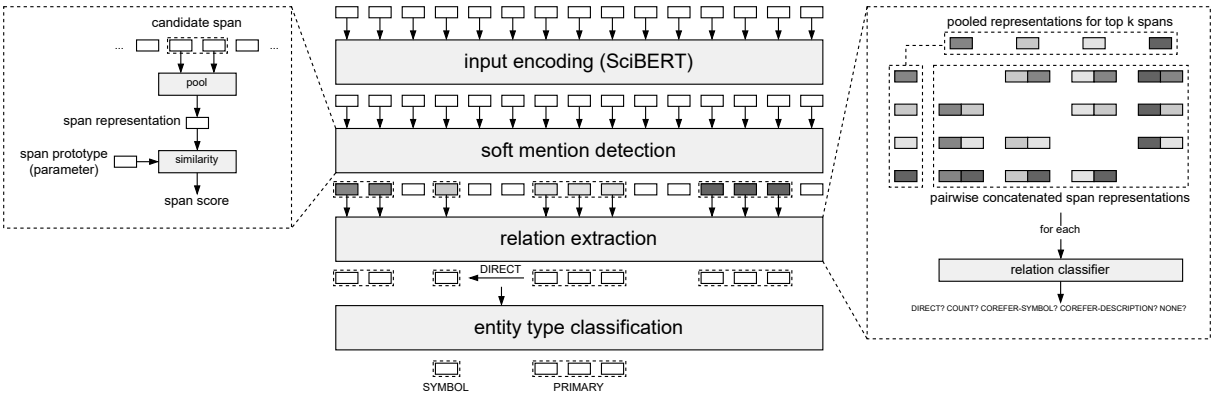

Figure 1: Architecture overview with detail illustrations for the soft mention detection (left) and relation extraction (right) modules. The layout of this figure was inspired by a similar figure found in (Eberts and Ulges, 2021).

2 Task Description

The task tackled in this paper is one of joint entity and relation extraction. This means, given an unannotated text as input, a system needs to (1) return annotations of relevant entity mention spans, (2) perform coreference resolution, (3) entity type classification, and finally (4) relation extraction on the identified spans. The specific task at hand has a number of key features that separate it from similar settings.

First, regarding entity extraction, the annotations and, thus, the final scoring are restricted to those entities which participate in relations. This means that a system which correctly identifies all symbols and descriptions in the input will score poorly even on the entity extraction portion of the final benchmark if the relation extraction is incorrect. More importantly from an engineering perspective, the resulting span annotations are incomplete in that they only include a partial set of valid spans for each document. In the entity extraction step we can, therefore, only reliably identify true positives and false negatives, not, however, false positives and true negatives.

Second, while coreference resolution (i.e., the linking of multiple mentions to a single entity) is part of the task, relation extraction is to be performed on a mention-level rather than the entity-level. This means that although a system may correctly identify a text span as being the description of a certain symbol, this classification will only be deemed correct in the evaluation if linked to the correct mention of said symbol. As a result, coreference links are interpreted as relations between mentions and thereby as part of the relation extraction subtask, rather than as part of the entity extraction subtask.

Third, entity types can be reliably inferred from the relations between them, meaning that instances of relations are only found between certain entity types. This feature can be used to inform the design of a system in two ways: Either, the task of relation extraction can be simplified by reducing the choices given to a classifier based on the entity types of two spans (i.e., a symbol cannot be the description to another symbol, therefore any such prediction can be disregarded), or the entity type classification can be informed by the relation extraction (i.e., if we identify a span A as the description of another span B, span A must be a description, while span B must be a symbol).

3 Approach

We propose an end-to-end entity and relation extraction system using a transformer-based language model, as illustrated in figure 1. The system consists of 4 modules: (1) The *input encoding module* tokenizes the input text and produces contextualized embeddings for each token, (2) the *soft mention detection module* ranks possible token spans by the likelihood with which they contain an entity mention, (3) the *relation extraction module* extracts relations on a subset of the highest ranked spans from the previous step, and finally (4) the *entity type classification module* assigns entity types to spans based on the relations detected between them.

3.1 Input Encoding

We examine two separate options of encoding the input: For the first option, we pass the input text to the language model without prior modification,

whereas for the second option, we perform preprocessing on the input to remove LaTeX code from the text portions of the input. Any input in LaTeX math mode is passed to the model unchanged.

Since our approach uses a transformer-based language model, the input needs to be tokenized. As a result of the tokenization, there are instances of relations which cannot be matched correctly by our model, due to the annotated span boundaries being contained within a token. For the training and development sets, this occurs in 1.99% and 2.84% of relation instances, respectively, and in these cases we adjust the labels accordingly.

3.2 Soft Mention Detection

Given that we cannot reliably identify false positives and true negatives from our labeled data, a mention detection strategy based on cross-entropy loss cannot be used for this task. Instead of following previous approaches in using feed-forward neural networks (Luan et al., 2018; Wadden et al., 2019), we propose a linear similarity based approach which ranks possible spans based on their similarity to multiple *prototype* embeddings (one prototype per entity type).

We begin by computing the set of all possible continuous spans up to a maximum length n and produce a fixed-size embedding e_s for each span by pooling the contextualized embeddings of all tokens within it. As pooling strategies we use either mean or max pooling. For each span embedding e_s we compute a span score X_s:

$$X_s = \max_{a_i \in A}(sim(e_s, a_i)) \quad (1)$$

where A is the set of prototype embeddings which contains an embedding for each entity type and $sim(a, b)$ is the dot product similarity of two vectors. We select the k spans with the highest values for X_s as our *candidate mentions* M for relation extraction. We compute the *mention loss* as the mean triplet loss (Schroff et al., 2015) across all prototype embeddings in A and all mentions in M.

3.3 Relation Extraction

For relation extraction, we use the document-level relation extraction model DL-MNAV (Popovic and Färber, 2022). We use the concatenation of two span representations as a representation for the relation between them (Wang et al., 2019). The resulting relation representations are compared to a single relation prototype embedding per relation type, as well as m additional prototypes representing the none-of-the-above class (this follows the MNAV model (Sabo et al., 2021)). The relation type corresponding to the prototype resulting in the highest dot product similarity for a relation representation is used as the predicted type. As loss function for the relation classification we use *adaptive thresholding loss* (Zhou et al., 2021) as it is capable of handling the large imbalance between positive and negative training examples present in document-level relation extraction tasks.

Due to quadratic scaling of the pairwise comparisons it is not feasible to perform relation extraction on all possible continuous spans. We, therefore, perform relation classification on the top k spans[2] with the highest span scores, meaning that we have to classify a maximum of $k(k \quad 1)$ relation representations for a given input text. The computational complexity of the system can, therefore, be adjusted dynamically at inference time by changing k, for example to be run on GPUs with smaller memory capacity or on GPUs with higher memory capacity to improve the quality of predictions.

As a result of the soft mention detection, it is possible that some of the k spans are overlapping and correspond to the same target (see appendix A.2 for examples). This means that the relation classifier may output multiple predictions for the same relation instance with slightly different mention spans. For predictions in which both the head and tail entity overlap, we therefore output only the prediction with the highest classification score.

3.4 Entity Type Classification

Finally, we use a simple mapping to determine the entity type of the spans which participate in the relations predicted by the relation classifier. The mapping used can be found in appendix A.1. For spans classified as "PRIMARY" we additionally change the predicted type to "ORDERED", if they are the head entity of more than one "Direct" relation.

4 Experimental Setup

For our language model we use SciBERT (Beltagy et al., 2019), which is trained on scientific

[2]During training we add annotated spans which are not among the top k spans.

		Entity Extraction								Relation Extraction					
pooling	preprocessing	F_1 strict [%]		F_1 exact [%]		F_1 partial [%]		F_1 type [%]		precision [%]		recall [%]		F_1 [%]	
Development set															
max	None	59.79	0.99	60.19	0.99	69.20	0.68	67.45	1.05	65.86	1.34	44.14	0.97	52.86	1.10
mean	None	58.02	3.67	58.49	3.70	69.14	2.02	66.98	2.31	61.27	5.09	44.58	1.56	51.61	2.87
max	LaTeX2Text	58.90	0.79	59.30	0.87	68.69	1.00	66.88	1.09	64.54	2.61	43.77	1.76	51.66	0.77
mean	LaTeX2Text	54.59	15.07	54.99	14.44	65.62	11.28	64.22	14.22	63.89	33.24	40.59	15.90	49.64	23.63
Test set															
max	None	-		-		37.83	0.85	37.88	0.85	45.80	5.80	20.96	0.08	28.66	1.19
mean	None	-		-		41.21	1.18	41.23	1.19	42.25	3.19	26.55	1.19	32.28	0.20
max	LaTeX2Text	-		-		38.33	1.57	38.38	1.57	46.09	0.77	21.64	1.60	29.45	1.41
mean	LaTeX2Text	-		-		34.53	11.02	34.64	11.13	47.02	20.70	18.20	8.10	26.24	11.64

Table 1: Entity and relation extraction scores for 4 different models on both the development and the test set. NER metrics strict and exact were not produced by the test set evaluation script on the competition site and the test set is not publicly available at the time of writing.

text, via Huggingface's Transformers library (Wolf et al., 2020). For LaTeX preprocessing (see section 3.1) we use Pylatexenc.[3] As our optimizer, we use AdamW (Loshchilov and Hutter, 2019) with learning rates $2 \cdot [3e{-}5; 5e{-}5; 7e{-}5]$, a linear warmup of 1 epoch followed by a linear decay to zero, for a total of 60 epochs,[4] a batch size of 4, and apply gradient clipping with a max norm of 1. During training, we randomly downsample the amount of candidate spans for soft mention detection to 1000, while ensuring that all labeled spans are included. During training and development set evaluation, we set k, the number of spans to perform relation classification on, to 50, as preliminary experiments showed this value to yield a good compromise between model performance and training time. For test set evaluation we increase k to 400. Training takes approximately 10 hours on a single NVIDIA V100 GPU using mixed precision. We perform early stopping based on the micro F_1 score for relation extraction on the development set. We train each hyperparameter configuration 3 times using different random seeds and report the median and standard deviation for each metric. As a result of the different combinations of preprocessing and mean-/max-pooling, we examine the performance of 4 configurations on the test set. For our evaluation, we report the micro F_1 scores for NER metrics as used in SemEval-2013 Task 9.1 (Segura-Bedmar et al., 2013)[5]. For relation extraction we report micro precision, recall and F_1 scores, unless otherwise indicated.

[3] https://github.com/phfaist/pylatexenc

[4] The length of one epoch is dictated by the number of training examples, which is 3119.

[5] We use the following implementation: https://github.com/davidsbatista/NER-Evaluation

5 Results

5.1 Overview

The results of the 4 model configurations on the test set are reported in table 1. In comparison to the other approaches taking part in SemEval-2022 Task 12, our system ranks in place 3/9 in terms of relation extraction F_1 score.[6]

In general, we find that our model produces predictions with significantly higher precision than recall.

5.2 Impact of Preprocessing

With respect to the preprocessing procedure, we observe no clear performance impact. We conclude that SciBERT appears to cope well with LaTeX code and preprocessing, as described in this paper, is not required.

5.3 Impact of Pooling Procedure

Regarding the pooling procedures we find that mean pooling tends to cause higher variability in the classification performance of the models. For the models trained using mean pooling and preprocessing, 1 of 3 models performed significantly worse than the others, causing the large standard deviation in the results.

5.4 Impact of Domain

In table 2, we show the relation extraction F_1 scores for a model across the 4 different domains covered by the development set paired with the distribution of training data across domains. We observe large performance differences depending on the domain with math and physics showing very high macro

[6] Scores for other metrics are not publicly visible on the leaderboard at the time of writing.

	domain			
	cs	econ	math	physics
% of training corpus	16.63	27.08	12.82	32.08
relation type				
Direct	33.12	21.05	63.82	85.71
Count	-	-	84.62	100.00
Corefer-Symbol	21.05	20.47	91.30	100.00
Corefer-Description	3.51	0.00	76.92	96.00
macro	19.23	13.84	79.17	95.43
micro	25.93	19.49	78.77	88.77

Table 2: F1 scores for relation extraction across different domains and relation types on the development set. cs and econ do not contain any instances of "Count".

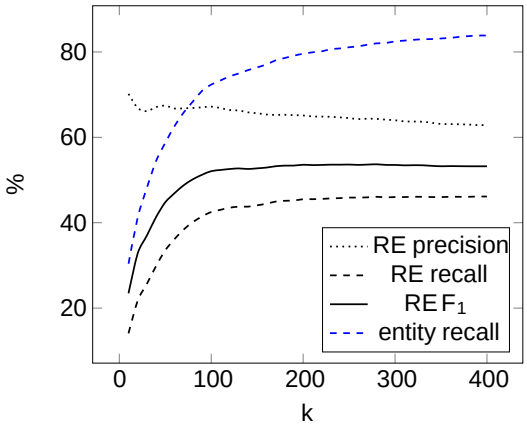

Figure 2: Plot of the impact of increasing values of k on precision, recall, and F1 scores on the development set.

F_1 scores (79.17% / 95.43%) and computer science and economics performing poorly (19.23% / 13.84%). While physics content does represent the majority of training examples, the distribution of domains across training examples does not fully explain the disparity.

5.5 Impact of k

In figure 2, we show the change in relation extraction performance across different values of k. We also include in the plot the percentage of entity spans in the top k ranked spans (entity recall). While the relation extraction performance improves proportional to the entity recall for $k < 100$ the improvement slows down for higher k. We hypothesize that this is due to the limiting of $k = 50$ and the candidate span downsampling during training, which prevents the model from seeing some of the more difficult cases. In appendix A.2, we show examples of detected spans.

matching	precision	recall	$F_{1;micro}$
strict	55.85	44.01	49.23
partial	62.87	46.13	53.22

Table 3: Comparison of strict and partial matching requirements with respect to classification scores on the development set.

5.6 Impact of Tokenization

In order to measure the impact of tokenization errors produced by adjusting labels during training, we perform a partial matching of relation labels as follows: For predicted relation triples which are false positives, we accept them as true positives for an annotated instance if the intersection-over-union (IOU) scores of both head and tail entities are greater than 67% and the predicted relation type matches the label. In table 3 we show the results of both strict and partial matching for our best model on the development set. We find that the relaxed requirements for span accuracy result in an increase in the F_1 score of 3.99%. We conclude that tokenization errors, while measurable, do not account for the majority of errors of our model.

6 Conclusion

In this paper, we present an end-to-end joint entity and relation extraction approach for linking mathematical symbols to their descriptions in LaTeX documents. Our model appears to be sensitive to the domain of the input documents, achieving high macro F_1 scores of 95.43% and 79.17% for physics and math content, respectively, while achieving macro F_1 scores of only 19.23% and 13.84% for computer science and economics related content. We find that the model's predictions are higher in precision than in recall. We perform a detailed error analysis and identify cross-domain generalization as the most critical problem to tackle in future work.

Acknowledgements

The authors acknowledge support by the state of Baden-Württemberg through bwHPC.

References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. Expert Systems with Applications, 114:34–45.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A 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), pages 3615–3620, Hong Kong, China. 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.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of Frontiers in Artificial Intelligence and Applications, pages 2006–2013. IOS Press.

Markus Eberts and Adrian Ulges. 2021. An end-to-end model for entity-level relation extraction using multi-instance learning. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, pages 3650–3660. Association for Computational Linguistics.

Michael Färber. 2019. The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data. In Proceedings of the 18th International Semantic Web Conference, ISWC'19, pages 113–129.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers. In Proceedings of The 12th International Workshop on Semantic Evaluation, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

Viet Lai, Amir Pouran Ben Veyseh, Franck Dernoncourt, and Thien Huu Nguyen. 2022. Semeval 2022 task 12: Symlink: Linking mathematical symbols to their descriptions. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022). Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations 2019, page 18.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Dat Quoc Nguyen and Karin Verspoor. 2019. End-to-End Neural Relation Extraction Using Deep Biaffine Attention. In Advances in Information Retrieval, Lecture Notes in Computer Science, pages 729–738, Cham. Springer International Publishing.

Nicholas Popovic and Michael Färber. 2022. Few-Shot Document-Level Relation Extraction. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies

Ofer Sabo, Yanai Elazar, Yoav Goldberg, and Ido Dagan. 2021. Revisiting Few-shot Relation Classification: Evaluation Data and Classification Schemes. Transactions of the Association for Computational Linguistics, 9:691–706.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages 815–823. IEEE Computer Society.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 341–350, Atlanta, Georgia, USA. 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.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara

Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Appendix

A.1 Entity Type Classification Map

Table 4 shows the classification map used for determining entity types based on relations for SemEval-2022 Task 12.

Relation	head entity	tail entity
Direct	PRIMARY*	SYMBOL
Count	PRIMARY	SYMBOL
Corefer-Symbol	SYMBOL	SYMBOL
Corefer-Description	PRIMARY	PRIMARY

Table 4: Classification map for entity types based on relations in which the spans participate. *In a postprocessing step, entity types of spans which are the head entity of multiple "Direct" relations are adjusted to "ORDERED".

A.2 Examples of Spans Detected for Different Values of k

Examples of spans detected via soft mention detection are shown in figures 3, 4, 5, and 6.

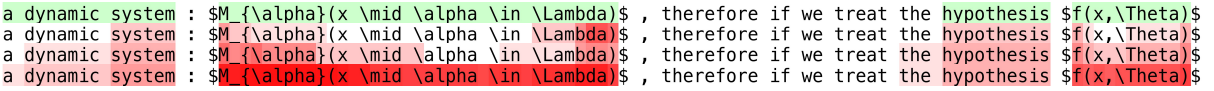

Figure 3: An example of spans detected in the domain of computer science. The top row shows ground truth labels in green, while the rows below are spans detected at $k = 50, 100, 150$.

Figure 4: An example of spans detected in the domain of economics. The top row shows ground truth labels in green, while the rows below are spans detected at $k = 50, 100, 150$.

Figure 5: An example of spans detected in the domain of mathematics. The top row shows ground truth labels in green, while the rows below are spans detected at $k = 50, 100, 150$.

Figure 6: An example of spans detected in the domain of physics. The top row shows ground truth labels in green, while the rows below are spans detected at $k = 50, 100, 150$.