NLP@VCU at BioASQ2025: Information Extraction on the GutBrainIE Dataset*

Notebook for the NLP at VCU Lab at CLEF 2025

Scott Taylor[1,*,†], Charlie Dil[1,*,†], Aaron Shah[1,*,†], Jannat[1], Cyd Oldham[1], Ayush Upadhyay[1], Joanne Varughese[1], Nicole Yazbeck[1] and Bridget T. McInnes[1,*]

[1]Department of Computer Science, Virginia Commonwealth University, 401 S. Main St., Rm E4222, Richmond, VA 23284, USA

Abstract

In this paper we present our solutions for both main subtasks of GutbrainIE @ CLEF 2025: Named Entity Recognition and Relationship Extraction on a corpus of biomedical text related to connections between microbiota, Parkinson's Disease and mental health [1]. Our NER approach uses Gliner-biomed to extract entities from the text, serving as a base for the predictions of our relationship extraction approaches. Our first two relationship extraction approaches used a DeBERTA-CNN based framework to effectively parse complex relationship extraction. Additionally, we implemented a Knowledge Hypergraph approach to research the capacity of hypergraphs to augment or supplement current LLM based methods. Our best performing system achieved third place in the Named Entity Recognition subtask with a micro-F1 score of 0.8370.

Keywords

Information Extraction, Named Entity Recognition, Relationship Extraction, Hypergraph Neural Network, HGNN, Knowledge Graph Representation

1. Introduction

This paper presents our system description in the participation of the BioASQ 2025 [2] Task 6 GutBrainIE Challenge [3]. Information Extraction (IE) can be broken down into two main tasks Named Entity Recognition (NER) and Relationship Extraction (RE). NER is a task that seeks to extract entities (such as persons, places, chemicals, or species) from text. NER often finds itself upstream of other NLP tasks, such as RE, entity linking, and coreference resolution. Relationship Extraction (RE) is the task of determining the relationships between entities. The GutBrainIE Subtask 6.1 of BioASQ at CLEF2025 uses NER as an upstream task for Relationship Extraction (RE).

In the GutBrainIE 2025 Subtask 6.2, there are three levels of RE. The first level is Binary Tag-based Relationship Extraction, where the task is to identify if there is a relation in a sample. The second level is Ternary Tag-based RE, where the goal is to identify not only if there is a relationship present, but also what that relationship type is. Finally, there is Ternary Mention-based RE, where the specific entity spans must also be extracted.

The paper is organized as follows: Section 2 describes the dataset; Section 3 describes our approaches; Section 4 presents our test set results and discusses approach performance, and Section 5 draws

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*You can use this document as the template for preparing your publication. We recommend using the latest version of the ceurart style.

*Corresponding author.

†These authors contributed equally.

‡**Author contributions:** Scott Taylor designed and implemented the NER model and provided writing and editing. Charlie Dil provided writing and editing, and designed, implemented, and supervised the implementation of the Deberta CNN RE models with support from Jannat and Nicole Yazbeck. Aaron Shah designed and implemented the Hypergraph Approach and provided writing and editing. Cyd Oldham, Joanne Varughese, and Ayush Upadhyay conducted NER research, with Cyd Oldham providing writing and editing. Bridget T. McInnes supervised the project.

taylorsm9@vcu.edu (S. Taylor); ndil@vcu.edu (C. Dil); shaham7@vcu.edu (A. Shah); lnuj3@vcu.edu (Jannat); oldhamc@vcu.edu (C. Oldham); upadhyaya@vcu.edu (A. Upadhyay); varughesej@vcu.edu (J. Varughese); yazbeckn@vcu.edu (N. Yazbeck); btmcinnes@vcu.edu (B. T. McInnes)

conclusions and outlines possible future work.

2. Dataset

The dataset used in the CLEF2025 Task 6 - GutBrainIE Challenge [3] is made from a collection of biomedical article titles and abstracts sourced from PubMed. These texts are centered on research exploring the relationship between the gut microbiome and the brain, with an emphasis on its relevance to neurological and psychiatric conditions. The data reflects findings from microbiology, neuroscience, psychiatry, and gastrointestinal research, providing a broad foundation for natural language processing tasks such as NER and RE.

The datasets are categorized into 4 quality tiers reflecting varying levels of expertise in the annotation curation. The highest quality, referred to as the platinum-standard annotations, were curated by experts within the CLEF2025 GutBrainIE Challenge organizing group and subsequently reviewed by biomedical specialists external to the core team. The gold-standard annotations were produced by experts as well, but without the additional external review that characterizes the platinum tier. The silver-standard annotations are considered intermediate quality and were generated by trained students working under expert supervision. Finally, the bronze-standard annotations represent the lowest quality level, with NER annotations produced automatically using a fine-tuned GLiNER[4] model and RE annotations generated by a fine-tuned ATLOP[5] model .

3. Methodology

3.1. Named Entity Recognition

All of our NER models use a version of Gliner Biomed[6], a variant of Gliner[4] that uses a deBERTa V3[7] backbone and is pretrained on synthetic data from the biomedical domain. We employed both gliner-biomed-large-v1.0 and gliner-biomed-bi-large-v1.0.

3.1.1. Loss Computation

For loss computation, we employ Gliner Masking[8] with the goal of mitigating incorrect learning from unlabeled entities in lower tiers of data. Using the library's global_wo_threshold, spans which are labeled as non-entities will be selectively masked during training. Specifically, the model's probability of these spans corresponding to an entity type is calculated and suspected unlabeled entities are probabilistically masked using Bernoulli sampling. Additionally, we employed focal loss to bias the model's learning towards difficult, misclassified samples. We use a mean loss reduction for all models.

3.1.2. Postprocessing

The default behavior of Gliner is to output predictions that are above a user selected certainty threshold. Rather than select a performant threshold, we output any prediction with a certainty of .01 or higher and learn a set of per-class certainty thresholds by maximizing model performance on the validation set using optuna. A separate set of class thresholds is learned for each model. These thresholds are applied as a post-processing step when doing inference on the test set, before any model's predictions are combined into an ensemble. During this step, we borrow additional postprocessing from the GutBrainIE organizers[9], modifying their method of combining successive gliner predictions to concatenate predictions of the same class that are separated by one character to find that character in the text, rather than using a space. This slightly improved our results on the validation set.

Post-processing rules were created following an error analysis of partially predicted spans on the validation set with model 1. Of the possible rules identified, two were found to help performance on the validation set: extending drug entities whose next word is "treatment" or "treatments", and

extending any entity type whose next word is "intervention" or "interventions". When predicting with an ensemble, these rules are applied after the ensemble predictions are merged.

3.1.3. Ensemble Model

Our ensemble is a simple rule-based ensemble[10] made by combining the outputs of three separate models. Model 1 serves as the base model, and the predictions are selectively supplemented or replaced by those of models 2 and 3 based on heuristic rules, discussed in section 3.1.7. Following Gliner[4] and Sainz et al.[11], we shuffle entity order and randomly drop entities as a regularization method. All models are trained using a cosine scheduler, weight decay for encoder parameters set to .1, weight decay for other parameters set to .05, dropout set to .4, and gradient clipping with maximum L2 norm of the gradient vector set to 10.

3.1.4. Model 1

Model 1 uses gliner-biomed-large-v1.0 and is trained on the GutBrainIE platinum, gold, silver, and bronze datasets for 20,000 steps. We set the learning rate of encoder parameters to 1e-5, and that of the other parameters to 5e-5. For focal loss, we use a gamma of 2 and an alpha of .75.

3.1.5. Model 2

Model 2 uses gliner-biomed-bi-large-v1.0 and is trained in a 2 stage process, first on BC5CDR[12] for 15,000 steps, and then on the GutBrainIE platinum, gold, and silver datasets for 5,000 steps. The goal of this 2 stage process was to leverage the large set of "disease" entities in the BC5CDR dataset to change the encoder representations for entities that may be present in GutBrainIE's largest class, "DDF". During the initial stage, the learning rate for encoder parameters was 1e-5, and 3e-5 for other parameters, with focal loss gamma set to 2 and alpha set to .75. For the second stage, the learning rate for other parameters was increased to 5e-5 and alpha decreased to .5.

3.1.6. Model 3

Model 3 uses gliner-biomed-bi-large-v1.0. Training data and hyperparameters are identical to those of Model 1, but with focal loss alpha decreased to .5.

3.1.7. Ensemble Predictions

While Model 1 had the best overall performance, Model 3 was found to have higher overall performance on the "anatomical location", "animal", and "human" classes. We replaced Model 1's predictions from these classes with those of Model 3. Model 2 was found to predict "microbiome" and "statistical technique" spans with higher overall performance than model 1, and we made the same type of replacement. Additionally, we identified classes that model 2 predicted with a higher precision than model 1: "DDF", "biomedical technique", and "statistical technique". For these classes, we retained model 1's higher recall predictions, but added the predictions of model 2, overwriting any model 1 predictions with overlapping spans.

3.2. Relationship Extraction

3.2.1. Approach 1

This model uses the platinum, gold, and silver training datasets from the GutBrainIE 2025 task dataset. For preprocessing, we first pull sentences with labeled relations. In the case of cross-sentence relations, i.e. relations that span multiple sentences (not necessarily consecutively) we concatenate the sentences which contain an entity that is in the relationship. Samples with no relation are created by sampling sentences with entity pairs that have the entity types of a defined relation. For example, "Anatomical

Location" and "Human" can form the "Located In" relation, so we sample "Anatomical Location" and "Human" pairs that do not have a relation. For every entity pair that is in a relation in the training data, there is an entity pair of the same types that is not in a relation, when possible. This process is done for platinum, gold, and silver separately, with the results of each combined.

We train a Convolutional Neural Network (CNN) using *deBERTa-v3-large* [13] [7] contextualized embeddings. After some preliminary analysis of the sample lengths, a maximum length of 500 was selected. The samples are fed into the model to get contextualized embeddings, to which we apply a 1D Convolutional Layer. The kernel size we use is 3, which projects the 768 dimensions into 256 dimensions per subword. After this, a ReLU activation function is applied. Then, to combat the potential for the same sentences having multiple relations, we specifically extract the representations for the two entities and concatenate them together. If the entity consists of multiple subwords, those subword representations are averaged together. This results in a vector of length $256 * 2$, which is passed into a linear layer which maps these representations to class predictions, which is an 18-dimensional vector where the index of the maximum logit is the prediction for the sample.

We use a starting learning rate of 5e-5 and a Cosine Annealing Learning Rate Scheduler. We also use Early Stopping based on Micro-F1

3.2.2. Approach 2

This method uses the same model architecture as the previous section with an additional preprocessing pruning step. After creating the sample datasets as described previously, less populated relationship types in platinum are fed into the deBERTa to get averaged vector representations of the sentences. Then, we average those sentence vectors per relationship type to get relation type vectors. Next, cosine similarity of gold samples that are part of the less populated relationship types in platinum and the relation type vector it corresponds with determines whether we keep the sample. Specifically, if the similarity is above or equal to 0.75, we add the gold sample to the platinum samples, otherwise, we drop the sample.

Table 1
Label Frequencies with Approach 1

Label	Frequency
target	2091
impact	1032
change effect	496
administered	72
located in	1525
is linked to	2241
influence	3088
affect	790
change abundance	674
part of	427
is a	81
interact	384
change expression	141
used by	589
compared to	17
produced by	260
strike	157
NONE	13015

Table 2
Label Frequencies with Approach 2

Label	Frequency
target	216
impact	105
change effect	27
administered	38
located in	177
is linked to	214
influence	249
affect	131
change abundance	69
part of	55
is a	63
interact	31
change expression	45
used by	102
compared to	7
produced by	7
strike	19
NONE	1317

3.2.3. Approach 3

Traditional RE frameworks typically treat entity pairs in isolation, learning local context representations per sentence or span. In contrast, our hypergraph formulation explicitly models higher-order interdependencies among entities at the type level, leveraging global co-occurrence and relational priors across an entire corpus or abstract.

Formally, we define a **typed relational hypergraph** as a tuple $\mathcal{H} = (V, E, X, Y)$, where:

- V is the set of nodes, each corresponding to an *entity type* (e.g., `Chemical`, `Anatomical Location`, `Microbe`),
- $X \in \mathbb{R}^{|V| \times d}$ is the node feature matrix, where each row is a 768-dimensional vector constructed by mean-pooling up to 250 contextual embeddings derived from BioBERT [14, 15],
- E is the set of *directed hyperedges*, each representing an ordered $(\texttt{subject\_type}, \texttt{object\_type})$ pair,
- $Y \in \mathbb{N}^{|E|}$ is the relation label vector, where each $y_i \in \mathcal{R}$ is one of 17 labeled relation types or NONE.

This hypergraph structure is **type-centric** rather than instance-centric: it encodes generalized interaction rules between types, abstracting away from individual mentions. While each edge is a directed pair $(\texttt{subject\_type}, \texttt{object\_type})$ associated with a relation label, the structure facilitates efficient parameter sharing and semantic generalization across abstracts. The resulting model is thus better suited to infer unseen relations by leveraging structural regularities across the entire corpus.

Although our edge structure is pairwise, we leverage `HypergraphConv` layers [16] from PyTorch Geometric [17] to enable multi-hop propagation over this relation graph. This propagation mechanism enriches type embeddings and allows abstract-level reasoning. This design reflects prior work in hypergraph neural architectures [18, 19, 20] and complements traditional span-level RE pipelines.

Unlike CNN-based architectures that operate over localized sentence windows, our hypergraph allows for **multi-hop diffusion of information** across type nodes. This enables the model to capture patterns such as:

- "Microbes that influence chemicals often also impact symptoms."
- "Entities of type X rarely participate in more than one relation in abstract Y."

The graph propagation mechanism is implemented using stacked `HypergraphConv` layers from PyTorch Geometric [17], which have been shown to effectively model higher-order interactions in prior work [16, 18, 19, 20]. This formulation introduces an inductive bias over the relation structure that complements traditional span-level classifiers and allows for abstract-wide reasoning.

By using global entity-type nodes rather than localized entity mentions, and performing learning on these symbolic hypergraphs, we align with trends in literature-based discovery and type-level generalization in biomedical NLP [21]. This approach improves both generalization to unseen documents and robustness to annotation sparsity, particularly in low-resource biomedical relation types.

3.2.4. Draft Hypergraph Construction

For training, we build a supervision hypergraph by enumerating every annotated relation instance from the gold, silver and bronze datasets. Each (subject–type, object–type, predicate) triple becomes a positive sample; duplicates and differing labels for the same type-pair are retained as distinct examples. We then sample an equal number of negative pairs uniformly from all remaining type-pairs (i $\neq$ j) to produce a 1:1 balance. These positives and negatives form the incidence matrix H_{inc}: each pair corresponds to an edge index, and the associated label tensor y drives a multiclass cross-entropy loss (with "no_relation" down-weighted to 0.35) [20, 18, 16].

At inference time, we construct a separate candidate hypergraph per abstract using the baseline NER output. We load the ensemble-plus-rules predictions, canonicalize the entity labels to the 13 allowed types, and materialize every ordered pair of distinct types as a directed edge. This dense candidate

graph, distinct from the training hypergraph, is used to enumerate all possible relation hypotheses and ensure maximal recall before refinement [20].

3.2.5. Hypergraph Neural Network Refinement

Node features are initialized by mean-pooling up to 250 BioBERT-base contextual embeddings per entity type, yielding a fixed 768-dimensional vector per node [14, 15]. The refinement model, `HGPairModel`, registers these embeddings as a buffer and applies an `HGStack` of four residual `HypergraphConv` layers (each followed by LayerNorm, ReLU and 10% dropout) to propagate information through H_{inc} [16, 19, 17]. For each candidate edge (u, v), we compute a feature vector

$$\left[h_u,\ h_v,\ h_u \odot h_v,\ |h_u - h_v|,\ \cos(h_u, h_v)\right] \ \in\ \mathbb{R}^{4\cdot 768+1},$$

which is fed into a three-layer MLP ($512 \to 256 \to |\text{RELATIONS}|$) to produce refined logits [21].

We train two instances (seeds 0 and 1) for 1601 steps each using AdamW (LR = 2×10^{-3}, weight decay = 5×10^{-4}) [22] with a `CosineAnnealingLR` schedule ($T_{\max} = 1600$) and weighted cross-entropy loss. Ensemble averaging of the two models' softmax outputs reduces variance. At inference, we average the two models' softmax outputs and retain only edges whose maximum confidence exceeds $\tau = 0.35$, replacing each placeholder label with its refined argmax. This deep, multi-hop convolutional stack combined with rich pair features and ensemble design (hopefully) drives improvements in both precision and recall [17, 20].

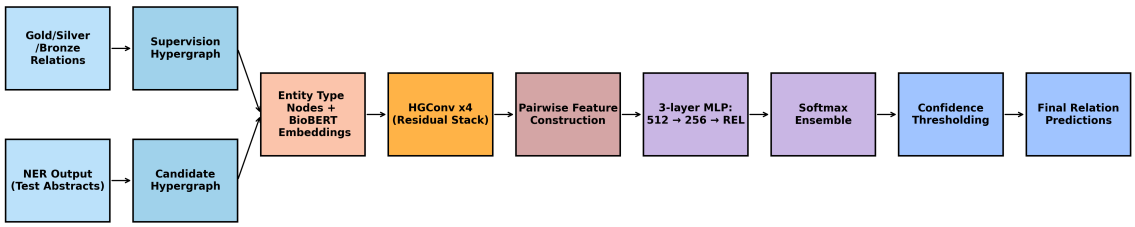

Figure 1: Visual representation of the hypergraph neural network prediction process

4. Results & Discussion

4.1. Evaluation Metrics

We report standard metrics for RE: precision (P), recall (R), and F1 score (F1), each computed in both macro and micro variants. The **macro-F1** is the unweighted mean of F1 scores computed per class, emphasizing performance on rare classes. The **micro-F1** aggregates over all relation predictions and is sensitive to class imbalance. We use the models provided by the GutBrainIE organizers[9] for baseline comparisons.

4.2. Subtask 6.1 - Named Entity Recognition (NER) Results

In Table 3, we examine the performance of our submitted ensemble, Ensemble1, on the GutBrainIE Test data for Subtask 6.1 - Named Entity Extraction (NER), using the provided baseline system for comparison [3].

We note the relatively low macro recall and macro F1 scores of our model. Training procedures and hyperparameters were selected based on their effect in maximizing the contest's reference metric for the leaderboard score: Micro F1. Due to the unbalanced class representation in the dataset, we chose to maximize this score through a greater focus on majority classes, at the partial expense of our performance on underrepresented classes.

Table 3
Performance of Ensemble 1 on Subtask 6.1 - Named Entity Recognition (NER)

System	Macro P	Macro R	Macro F1	Micro P	Micro R	Micro F1
Baseline	0.6883	**0.7690**	0.7047	0.7639	0.8238	0.7927
Ensemble1	**0.8139**	0.7161	**0.7169**	**0.8255**	**0.8488**	**0.8370**

4.3. Subtask 6.2 - Relation Extraction (RE) deBERTa CNN Results

In Table 4, we evaluate the performance of the deBERTa CNN approaches on the GutBrainIE Test data, using the provided baseline system for comparison [3].

Table 4
Performance of deBERTa-CNN approaches on Subtask 6.2 - Relation Extraction (RE)

Subtask	Approach	Macro Prec.	Macro Rec.	Macro F1	Micro Prec.	Micro Rec.	Micro F1
6.2.1	Baseline	**0.465**	0.356	0.386	**0.758**	0.489	**0.595**
6.2.1	Approach 1	0.397	**0.842**	**0.508**	0.438	**0.857**	0.580
6.2.1	Approach 2	0.339	0.663	0.426	0.441	0.792	0.567
6.2.2	Baseline	**0.473**	0.342	0.375	**0.753**	0.465	**0.575**
6.2.2	Approach 1	0.381	**0.800**	**0.487**	0.436	**0.844**	0.575
6.2.2	Approach 2	0.315	0.616	0.393	0.435	0.770	0.556
6.2.3	Baseline	**0.351**	0.183	**0.212**	**0.499**	0.245	**0.329**
6.2.3	Approach 1	0.137	**0.481**	0.197	0.111	**0.583**	0.187
6.2.3	Approach 2	0.129	0.453	0.188	0.122	0.564	0.201

We note that for Subtasks 6.2.1 and 6.2.2, Approach 1 (training on the Platinum, Gold, and Silver training data) achieves a higher Micro-F1 than Approach 2 (training on the Platinum and pruned Gold training data). Approach 1 scores 0.580 and 0.575 on Subtasks 6.2.1 and 6.2.2, whereas Approach 2 scores 0.567 and 0.556 on those Subtasks respectively. However, for Subtask 6.2.3, Approach 2 scores higher in terms of Micro-F1 than Approach 1, scoring 0.201 as opposed to 0.187. Overall for these two approaches, we observe a high recall and a low precision.

4.4. Subtask 6.2.1 - Binary Tag-based Relation Extraction (BT-RE) Hypergraph Results

Table 5 presents the performance of our hypergraph-based models on the Subtask 6.2.1 Binary Tag-based Relation Extraction (BT-RE). We include individual models (HGmodel4, HGmodel6) as well as ensemble variants (HGensemble1, 2, 3).

Table 5
Performance of Hypergraph Approach on Subtask 6.2.1 - BT-RE

System	Macro P	Macro R	Macro F1	Micro P	Micro R	Micro F1
baseline	**0.5181**	0.4330	**0.4404**	**0.6585**	0.4909	**0.5625**
HGensemble1	0.282	0.592	0.355	0.357	0.658	0.463
HGensemble2	0.305	0.603	0.379	0.368	0.645	0.469
HGensemble3	0.269	0.711	0.365	0.340	**0.736**	0.465
HGmodel4	0.297	0.570	0.366	0.366	0.645	0.467
HGmodel6	0.290	**0.718**	0.388	0.343	**0.736**	0.468

We observe that the ensemble models consistently outperform single models in terms of both macro-F1 and micro-F1, validating the effectiveness of ensemble averaging for improving generalization. The

strong recall of HGmodel6 (>71% macro) indicates that individual deep stacks can generalize well, though they may overfit or miscalibrate. Ensemble models like HGensemble2 deliver more balanced performance across precision and recall, suggesting that our ensemble averaging strategy meaningfully compensates for variance in training outcomes. These findings reinforce the value of combining deep hypergraph inference with robust, relation-aware pairwise features.

5. Conclusions and Future Work

With the NER ensemble, our overall performance was hurt by poor performance on minority classes. This could be improved by training expert models in some minority classes and either adding them to an ensemble directly, or using them to distill a single model using KL divergence loss.

In the RE with the deBERTa CNN approach, using Platinum, Gold, and Silver training data for Approach 1 yielded higher Micro-F1 when compared to using Platinum and pruned Gold training data for Approach 2. With both of these methods, we observed high recall and low precision, implying both of these methods yield many false positives. In the future, we plan to apply our pruning method to the Silver training data to determine whether increasing the number of instances improves performance over Approach 2 while also reducing noise from Approach 1.

Per our hypergraph based approaches, HGmodel6 achieved the highest macro recall and F1, HGensemble2 achieved the best micro-F1 overall. The consistent gains across ensemble models suggest reduced variance and improved stability. We plan to apply bootstrap resampling and McNemar's test in future work to establish statistical significance of observed differences between models.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. Rodríguez-Ortega, E. Rodriguez-López, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, D. Dimitriadis, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, M. Martinelli, G. Silvello, G. Paliouras, Overview of BioASQ 2025: The thirteenth BioASQ challenge on large-scale biomedical semantic indexing and question answering, volume TBA of *Lecture Notes in Computer Science*, Springer, 2025, p. TBA.

[2] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. R. Ortega, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, L. Menotti, G. Silvello, G. Paliouras, BioASQ at CLEF2025: The Thirteenth Edition of the Large-Scale Biomedical Semantic Indexing and Question Answering Challenge, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2025, pp. 407–415. doi:10.1007/978-3-031-88720-8_61.

[3] M. Martinelli, G. Silvello, V. Bonato, G. M. Di Nunzio, N. Ferro, O. Irrera, S. Marchesin, L. Menotti, F. Vezzani, Overview of GutBrainIE@CLEF 2025: Gut-Brain Interplay Information Extraction, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), CLEF 2025 Working Notes, 2025.

[4] U. Zaratiana, N. Tomeh, P. Holat, T. Charnois, Gliner: Generalist model for named entity recognition using bidirectional transformer, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 5364–5376. URL: https://aclanthology.org/2024.naacl-long.300/. doi:10.18653/v1/2024.naacl-long.300.

[5] W. Zhou, K. Huang, T. Ma, J. Huang, Document-level relation extraction with adaptive thresholding and localized context pooling, in: Proceedings of the AAAI Conference on Artificial Intelligence,

volume 35, 2021, pp. 14612–14620. URL: https://doi.org/10.1609/aaai.v35i16.17717. doi:10.1609/aaai.v35i16.17717.

[6] A. Yazdani, I. Stepanov, D. Teodoro, GLiNER-biomed: A Suite of Efficient Models for Open Biomedical Named Entity Recognition, 2025. URL: http://arxiv.org/abs/2504.00676. doi:10.48550/arXiv.2504.00676, arXiv:2504.00676 [cs].

[7] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. arXiv:2111.09543.

[8] sadpush, sabdoudaoura/GLiNER_masking, 2025. URL: https://github.com/sabdoudaoura/GLiNER_masking, original-date: 2025-02-14T08:42:13Z.

[9] M. Martinelli, MMartinelli-hub/GutBrainIE_2025_baseline, 2025. URL: https://github.com/MMartinelli-hub/GutBrainIE_2025_Baseline, original-date: 2025-03-05T15:28:27Z.

[10] S. Doan, N. Collier, H. Xu, P. H. Duy, T. M. Phuong, Recognition of medication information from discharge summaries using ensembles of classifiers, BMC Medical Informatics and Decision Making 12 (2012) 36. doi:10.1186/1472-6947-12-36.

[11] O. Sainz, I. García-Ferrero, R. Agerri, O. L. d. Lacalle, G. Rigau, E. Agirre, GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction, 2024. URL: http://arxiv.org/abs/2310.03668. doi:10.48550/arXiv.2310.03668, arXiv:2310.03668 [cs].

[12] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C.-H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, Z. Lu, BioCreative V CDR task corpus: a resource for chemical disease relation extraction, Database 2016 (2016) baw068. URL: https://doi.org/10.1093/database/baw068. doi:10.1093/database/baw068.

[13] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL: https://arxiv.org/abs/2006.03654. arXiv:2006.03654.

[14] J. Lee, W. Yoon, S. Kim, D. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240.

[15] H. Face, BioBERT: Pretrained biomedical language model, https://huggingface.co/dmis-lab/biobert-base-cased-v1.1, 2022. Accessed May 2025.

[16] Y. Bai, H. Zhang, P. H. S. Torr, S. Bai, Hypergraph convolution and hypergraph attention, arXiv preprint arXiv:1901.08150 (2021).

[17] M. Fey, J. E. Lenssen, Fast graph representation learning with pytorch geometric, arXiv preprint arXiv:1903.02428 (2019).

[18] N. Yadati, M. Nimishakavi, P. Yadav, M. Nimishakavi, P. Talukdar, Hypergcn: A new method for training graph convolutional networks on hypergraphs, arXiv preprint arXiv:1809.02589 (2019).

[19] E. K. Chien, Y. Peng, J. Xu, Z. S. Yu, Adaptive universal generalization bounds for hypergraph neural networks, arXiv preprint arXiv:1901.08746 (2019).

[20] W. Zhang, T. Gao, P. He, X. Liu, J. Gao, X. Wang, T. Liu, Learning to construct hypergraphs for text classification, arXiv preprint arXiv:2305.17386 (2023).

[21] Y. Yang, Y. Lin, Z. Liu, M. Sun, S. Wang, Learning to rank for biomedical entity normalization, arXiv preprint arXiv:2204.06584 (2022).

[22] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).