Interactive Span Recommendation for Biomedical Text

Louis Blankemeier
Stanford University
lblankem@stanford.edu

Theodore Zhao
Microsoft Health AI
theodorezhao@microsoft.com

Robert Tinn
Microsoft Health AI
robert.tinn@microsoft.com

Sid Kiblawi
Microsoft Health AI
sidkiblawi@microsoft.com

Yu Gu
Microsoft Health AI
aiden.gu@microsoft.com

Akshay S. Chaudhari
Stanford University
akshaysc@stanford.edu

Hoifung Poon
Microsoft Research
robert.tinn@microsoft.com

Sheng Zhang
Microsoft Research
zhang.sheng@microsoft.com

Mu Wei
Microsoft Health AI
muhsin.wei@microsoft.com

Joseph S. Preston
Microsoft Health AI
sam.preston@microsoft.com

Abstract

Motivated by the scarcity of high-quality labeled biomedical text, as well as the success of data programming, we introduce *KRISS-Search*. By leveraging the Unified Medical Language Systems (UMLS) ontology, KRISS-Search addresses an interactive few-shot span recommendation task that we propose. We first introduce *unsupervised KRISS-Search* and show that our method outperforms existing methods in identifying spans that are semantically similar to a given span of interest, with $>$ 50% AUPRC improvement relative to PubMedBERT. We then introduce *supervised KRISS-Search*, which leverages human interaction to improve the notion of similarity used by unsupervised KRISS-Search. Through simulated human feedback, we demonstrate an enhanced F1 score of 0.68 in classifying spans as semantically similar or different in the low-label setting, outperforming PubMedBERT by 2 F1 points. Finally, supervised KRISS-Search demonstrates competitive or superior performance compared to PubMedBERT in few-shot biomedical named entity recognition (NER) across five benchmark datasets, with an average improvement of 5.6 F1 points. We envision KRISS-Search increasing the efficiency of programmatic data labeling and also providing broader utility as an interactive biomedical search engine.

1 Introduction

One of the major challenges in developing machine learning models for biomedical text analysis is the scarcity of high-quality labeled data. Manual annotation of biomedical text is a time-consuming process that demands specialized expertise, leading researchers to investigate alternative methods such as weak supervision (Zhang et al., 2022a; Yakimovich et al., 2021; Poon et al., 2021; Lang and Poon, 2020) and active learning(Naseem et al., 2021; Ren et al., 2020) to address this bottleneck. Programmatic data labeling (Ratner et al., 2016, 2017b,a), a form of weak supervision in which domain experts develop heuristics (labeling functions) to provide noisy labels for large datasets, has been shown to be effective in leveraging domain expertise. However, developing diverse and high-quality labeling functions can be challenging, as it requires knowledge of the programmatic rule specification. Some techniques have been proposed to suggest labeling functions to users (Boecking et al., 2021; Zhao et al., 2021; Li et al., 2021), but they still rely on users' understanding of rule specifications to evaluate or modify the labeling functions.

To address this challenge, we introduce an interactive span recommendation task. Our key idea is to train a single model and adapt it to human feedback, enabling it to understand and treat similarity at various levels of granularity. This approach eliminates the need to train multiple models for different notions of similarity. Conventional entity linking is one such notion of similarity, where a user may want to identify all mentions of a specific concept, such as "hypertension disease" from the Unified Medical Language System (UMLS). However, a user may want the flexibility to iden-

Proceedings of the 5th Clinical Natural Language Processing Workshop, pages 373–384

tify not only mentions of "hypertension disease" but also those of "hypertension treatments" and "hypertension comorbidities" simultaneously. This task extends beyond entity linking and can be more broadly described as interactive span recommendation.

To tackle the interactive span recommendation task, we propose *KRISS-Search*, a method that enables domain experts to develop span recommendation models for searching unlabeled corpora. A crucial aspect of model performance in KRISS-Search is the choice of embedding space. The UMLS ontology offers a comprehensive set of biomedical concepts organized hierarchically. We adapt the UMLS-based self-supervised training technique of KRISSBERT to generate the embedding space used by our method.

We evaluate two versions of KRISS-Search. *Unsupervised KRISS-Search* takes a single user-selected query span from a biomedical corpus as input and returns semantically similar spans. However, in some cases, this single measure of similarity may not adequately overcome the inherent ambiguity in identifying spans based on one example. To address this limitation, we introduce *supervised KRISS-Search*, which employs active learning to incorporate human feedback and refine the concept of similarity used in the unsupervised version. In the context of programmatic data labeling, we envision unsupervised KRISS-Search recommending terms for users to incorporate into labeling functions and supervised KRISS-Search directly generating noisy labels, providing a more flexible alternative to labeling functions.

Our main contributions can be summarized as follows:

1. We demonstrate that unsupervised KRISS-Search outperforms PubMedBERT (Gu et al., 2020) by 51% area under the precision-recall curve (AUPRC) in returning spans with *exact* concept unique identifier (CUI) matches to the CUI associated with the query span. KRISS-search further outperforms PubMedBERT by 54% in returning spans with *similar* associated CUIs.

2. By extending unsupervised KRISS-Search to supervised KRISS-Search through human-feedback and active learning, we surface spans associated with specific concepts (CUIs) with an F1 of 0.68, outperforming PubMedBERT by 2 F1 points.

3. We demonstrate that supervised KRISS-Search performs comparably or outperforms PubMedBERT across five benchmark tasks in the few-shot biomedical NER setting. On average, supervised KRISS-Search outperforms PubMedBERT by 5.6 F1 points, demonstrating the flexibility of our method to handle various levels of granularity.

2 Methods

In this paper, we compare various training strategies for the BERT-base (Devlin et al., 2018) (100 million parameters) architecture in order to address our proposed task. While the training strategies discussed in this paper are specific to the BERT-base architecture, they can also be applied to larger models. The methods we evaluate can be characterized as "contextual," "in-domain," "contrastive," and "interactive." "Contextual" methods use the surrounding context to make recommendations, while "in-domain" methods are trained on data specifically related to the biomedical domain. "Contrastive" methods utilize semantic similarity and dissimilarity during the training process. "Interactive" methods involve human participation to guide model training. The four training strategies we compare are BERT, PubMedBERT, unsupervised KRISS-Search, and supervised KRISS-Search. Each strategy implements an additional descriptor in the order they were listed, with supervised KRISS-Search implementing all four.

To highlight the distinctions between BERT, PubMedBERT, unsupervised KRISS-Search, and supervised KRISS-Search, consider the following example. In the sentence "The patient received a pt assay," the query span "pt" refers to the concept "prothrombin time assay". BERT, which is not specifically tailored to the biomedical domain or designed to employ contrastive or interactive techniques, may surface a false positive "platinum," which shares the same abbreviation "pt" but is not relevant to the biomedical domain. Similarly, PubMedBERT, which is trained on biomedical data but does not utilize contrastive learning, may generate a false positive "physical therapy," which is in the biomedical domain but semantically dissimilar to the query span. In contrast, both unsupervised and supervised KRISS-Search utilize contrastive learning, which makes them more likely to recommend semantically similar spans, such as "prothrombin time assay", as this similarity is explicitly incor-

porated into the training process. Now consider another example: "decrease in right lung mass compared to prior imaging". Here, the user is interested in the query span "decrease in right lung mass", which represents a relationship between "decrease" and "right lung mass". In this scenario, the concept of similarity is complex and may require the interactive feature of supervised KRISS-Search to surface similar spans.

2.1 Efficiently Embedding the Corpus

We posit that KRISSBERT (Zhang et al., 2022b) serves as an excellent foundation for our method, as it is trained using a contrastive learning approach based on the UMLS ontology that enables it to effectively predict the correspondence of multiple entities to the same underlying concept, a task known as entity-linking. However, in its original form, KRISSBERT is not computationally tractable for our use case. The KRISSBERT model (Zhang et al., 2022b) uses the [CLS] token to represent the contextual embedding of a span and places entity tokens between the span and its context to communicate the span of interest to the model. As such, generating embeddings for X spans requires X forward passes. This can prove computationally intractable when the number of spans to embed is large. To address this issue, the KRISS-Search method removes the entity tokens from the mention representations and instead aggregates the final layer embeddings of the tokens in a span to generate the span's embedding. Fig. 2a shows how KRISSBERT uses entity tokens (corresponding embeddings shown in orange) to denote the entity and [CLS] embeddings to compute the contrastive loss. Fig. 2b shows how KRISS-Search removes the entity tokens and aggregates the final layer embeddings of the entity tokens to compute the loss. The dummy text snippets in Fig. 2 provide an example of a positive pair where "patient discharge" and "released" correspond to the same concept and are thus pulled together in the embedding space during contrastive training. The entity encoder is left unchanged and is trained jointly with the mention encoder, as we hypothesize that the hierarchical UMLS ontology embedded in the entity encoder is useful for the task. For training, we used a single Tesla V100 16GB GPU.

These modifications increase computational efficiency by reducing the number of forward passes required for generating embeddings. If we pass 512

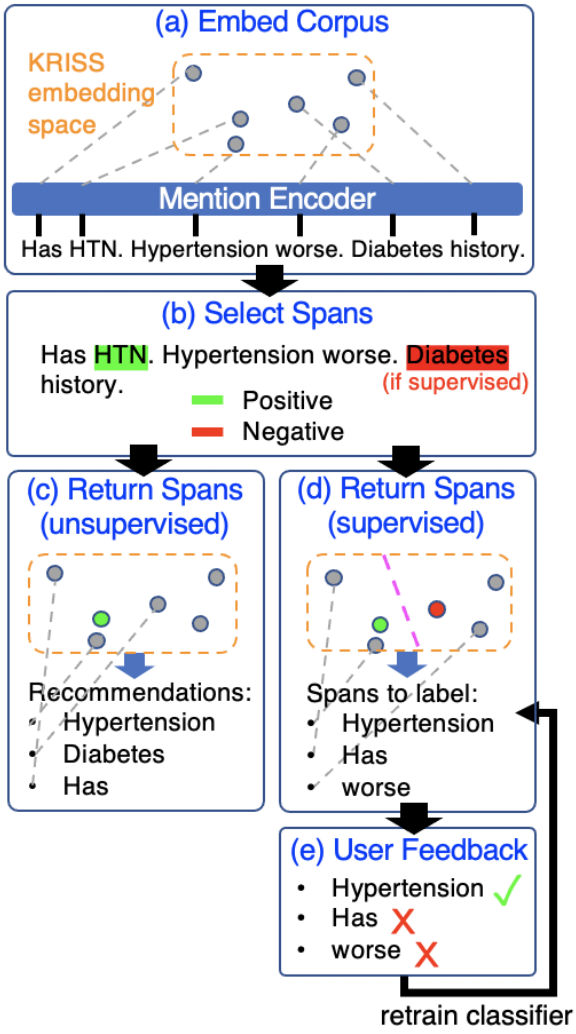
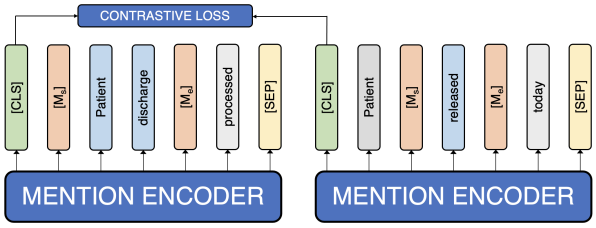
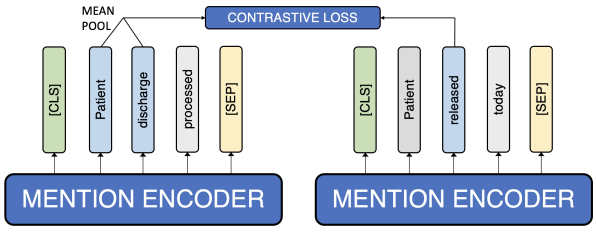


Figure 1: The KRISS-Search method consists of the following steps: (a) embed the corpus using the KRISSBERT embedding space, which places mentions of the same concept closer together and different concepts further apart; (b) the user selects spans to seed supervised and unsupervised KRISS-Search. For unsupervised KRISS-Search, the user selects a single positive query span. For supervised KRISS-Search, the user selects any number of positive and negative spans; (c) in unsupervised KRISS-Search, nearest neighbors to the positive query span are returned; (d) in supervised KRISS-Search, active learning is used to train a light-weight classifier to refine recommendations, with examples closest to the decision boundary being returned for subsequent active learning; (e) the user provides feedback on the returned spans, which can be used to retrain the light-weight classifier and return to step (d).



(a) KRISSBERT mention encoder training with entity tokens. The [CLS] token is used for computing the contrastive loss. This is the approach used in the original KRISSBERT paper.



(b) KRISSBERT mention encoder training without entity tokens. Span token embeddings (blue) are agggregated to generate the span embeddings and compute the contrastive loss. This is the strategy adopted for KRISS-Search.

Figure 2: A comparison of the mention encoder training with and without the entity tokens.

tokens (the maximum sequence length of BERT-base) into our model during a single forward pass, our method reduces inference time by N $\times$ 512 where N is the maximum span length that we embed. Additionally, our approach allows us to leverage the contrastive loss while still maintaining per-token embeddings. We use the same hyperparameters to retrain KRISSBERT and observe marginally degraded performance on validation data for the original KRISSBERT entity linking task. We note that this is expected as the KRISSBERT hyperparameters optimize validation performance of the original model. As the goal of this paper is not entity linking, we leave re-selecting hyperparameters to future work.

To further increase the efficiency of our method, we also filter the embeddings, discarding spans where the tokenization (Honnibal and Montani, 2017) of the span triggers a stop token, punctuation token, or whitespace token based on the assumption that such spans are not generally of interest.

2.2 Unsupervised KRISS-Search

The unsupervised KRISS-Search task involves returning a ranked list of spans from the corpus that are semantically similar to a query span, as determined by the L2 distance of their embedding to the query span embedding.

Evaluation: For evaluation of unsupervised KRISS-Search, we use the n2c2 dataset (2019 n2c2/UMass Lowell shared task 3) (Luo et al., 2020). This dataset contains 100 discharge summaries labeled with CUI annotations. We choose this dataset as it represents a domain shift from the PubMed abstracts used to train KRISSBERT. Additionally, n2c2 is annotated with diverse entities, including medical problems, treatments, and tests from established ontologies (Liu et al., 2005; Spackman et al., 1997).

To evaluate the quality of the retrieved spans, we assess the model's ability to retrieve (1) spans with associated CUIs that match the CUI associated with the query span (*same* evaluation type in Tables 1, 2, and 3) and (2) spans with associated CUIs that are closely related to the CUI associated with the query span (*related* evaluation type in Tables 1, 2, and 3). Related CUIs are generated by sampling a parent CUI of the query-associated CUI and returning its children using the UMLS hierarchy (Bodenreider, 2004). The *same* evaluation type experiments indicate how well each approach is at returning specific concepts of interest, while the *related* evaluation type experiments measure how well each approach can return more loosely related concepts.

We adopt a relaxed evaluation measure where spans that overlap with a concept mention are associated with the concept. We apply relaxed evaluation as we hypothesize that for our task, generating precise span boundaries is less important than providing the user with a greater number of recommendations. We represent spans with the mean of the span token embeddings. We choose the test query spans, used in Tables 1, 2, 3, as well as Figures 5 and 10, as follows. For 255 CUIs with more than 25 mentions in the corpus and corresponding span embeddings, we randomly sample one span for each of the 255 CUIs. We select CUIs that appear more than 25 times hypothesizing the difficulty of comparing approaches using low-prevalence CUIs.

To assess the model performance, we calculate the average precision, recall, and F1 metrics for a varying number of retrieved spans (Fig. 1 and Fig. 10). Specifically, we evaluate the performance at $1 \times N$, $2 \times N$, and $3 \times N$, where N represents the total number of mentions of a specific CUI in the dataset. It is important to note that N varies across different CUIs. The precision, recall, and F1 metrics are computed based on the number of correctly retrieved mentions of a specific CUI relative

to the total number of CUI mentions present in the dataset. The denominator of precision corresponds to the number of nearest neighbors retrieved, while the denominator of recall corresponds to the total number of mentions in the corpus for each CUI. These average values are not optimally informative as performance across different CUIs varies widely for all methods. As such, we also report per-query measures (Table 1). We compute average per-query percent recall improvement of KRISS-Search compared to PubMedBERT ($\overline{\%\Delta}$ in Table 1) and the frequency with which unsupervised KRISS-Search outperforms PubMedBERT with respect to recall ("Win Rate" in Table 1). We also compute p-values testing the null hypothesis that the means of the recalls from unsupervised KRISS-Search and PubMedBERT are the same using a two-sample t-test ("P-Val" in Table 1).

Additionally, we compute AUPRC values across the 255 test query spans for both the *same* and *related* experiments ($\overline{\text{AUPRC}}$ in Table 2). As with the recall measures, we compute average per-query percent AUPRC improvement of KRISS-Search compared to PubMedBERT ($\overline{\%\Delta}$ in Table 3), the frequency with which unsupervised KRISS-Search outperforms PubMedBERT with respect to AUPRC ("Win Rate" in Table 3), and p-values testing the null hypothesis that the means of the AUPRCs from unsupervised KRISS-Search and PubMedBERT are the same using a two-sample t-test ("P-Val" in Table 3).

2.3 Supervised KRISS-Search

To incorporate human feedback, we train a light-weight classifier with KRISSBERT embeddings as input. We cache the KRISSBERT embeddings to reduce the latency that would result from fine-tuning KRISSBERT and embedding the corpus at each active learning iteration. Our active learning strategy is as follows. First, the user selects a small number of positive and negative seed examples. We then train the light-weight classifier on these seed examples. Leveraging this trained model, we generate a small number of additional examples to be labeled and added to the training dataset. We then retrain the classifier from scratch, repeating this procedure until the label quality appears satisfactory.

Concept Retrieval: To measure the performance of supervised KRISS-Search in retrieving specific concept mentions, we use same 2019 n2c2 entity linking dataset that was used to evaluate unsupervised KRISS-Search. We simulate human feedback with the ground truth labels. We adopt a least confidence (LC in Table 5 and Table 4) active learning strategy where we return examples closest to the decision boundary for labeling. Furthermore, we use a logistic regression linear probe as the classifier, 5 active learning iterations, 15 seed examples, and 15 labeled examples per active learning iteration. Furthermore, we hypothesize that the contrastive loss makes distance to positively labeled examples a useful feature. Thus, we append the square of the L2 distance from the mean of the positively labeled embeddings to the KRISSBERT embeddings as an additional input feature, which we refer to as sum of squares (SS in Table 5 and Table 4). For these experiments, we use 28 concepts with greater than 100 mentions and corresponding embeddings, as we require additional spans for active learning. For evaluation, we compute performance on retrieving all ground truth mentions in the corpus.

Few-Shot Biomedical Named Entity Recognition: We evaluate our method on the BLURB NER datasets (Gu et al., 2020) to ground our method in benchmarked tasks and demonstrate the flexibility of our method to handle various notions of similarity. Here, we adopt strict evaluation as is conventional in NER and to be consistent with previous work evaluated on these tasks. We hypothesize that mean pooling aggregation does not sufficiently represent span boundaries, as it discards spatial information about span embeddings. Thus, we concatenate the first token embedding with the last token embedding and append the length of the span. To provide a fair comparison between the traditional NER approaches and KRISS-Search, we equalize the number of labeled words used for training. We empirically choose the total number of labeled words to be equal to the number of words in 75 randomly sampled sentences that are used for BERT and PubMedBERT training. For all methods, we use the same single layer perceptron as the light-weight classifier. During BERT and PubMedBERT training, we save training checkpoints, and for testing, we choose checkpoints with the best performance on the validation sets. We forgo this approach with KRISS-Search, as we assume that the user has not labeled validation sets. We report results (Table 4) using the random sampling baseline (RSB), least confidence active learning (LC), and a spatial refinement strategy (SpR).

2.4 KRISS-Search with Spatial Refinement

Supervised KRISS-Search is different from standard active learning tasks in that the examples (spans) are not independent, rather they have spatial relationships. Specifically, since one span can overlap with other spans in the sample set, we apply the following spatial refinement (SpR in Table 5, Table 4, and Fig. 7) strategies for KRISS-Search:

- When the span presented to the user overlaps with a true positive span, the user can modify the boundaries and label the correct span (Fig. 3).
- In NER tasks aiming for exact span recovery, only one span from an overlapping group of spans can be correct, in which case we predict only the span with highest probability and mark all the other spans as negative.

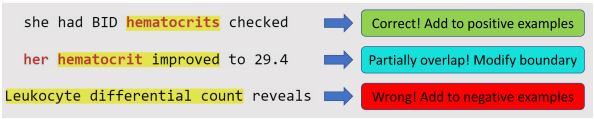


Figure 3: An example human feedback interface in Supervised KRISS-Search with spatial refinement (SpR). Yellow highlighting depicts spans presented to the user. Red bold letters are ground truth positive spans. For any recommended span, the user provides feedback by choosing from the following options: 1. mark the span as exactly correct (green button); 2. refine the boundaries of the span if it overlaps with a true span (cyan button); 3. mark the span as wrong (red button).

3 Results

3.1 Unsupervised KRISS-Search

Fig. 4 demonstrates a performant example on a test query for the "prothrombin time assay" CUI. Here, we show recall for unsupervised KRISS-Search (blue), PubMedBERT (red), and BERT (green) vs. the number of nearest neighbors for the *same* evaluation type. For this example, unsupervised KRISS-Search has an edge in terms of recall and thus precision, requiring fewer nearest neighbors to retrieve a similar number of positive spans. Fig. 8 in A demonstrates a similar outcome for this example using the *related* evaluation type.

Fig. 5 shows the mean recall, precision, and F1 across the 255 test query spans for the *same* evaluation type. Across the 255 corresponding concepts, an average (standard deviation) of 47%

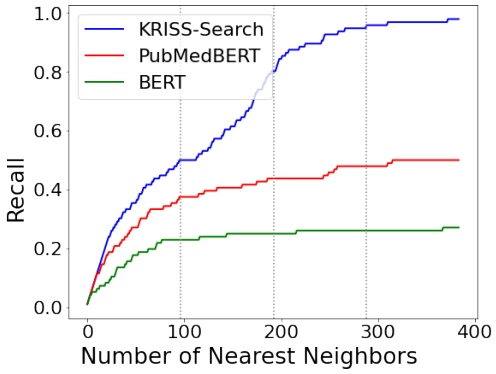


Figure 4: Recall using the *same* evaluation type (CUIs associated with returned spans must match the query associated CUI exactly). Query span is "PT", corresponding to the concept "prothrombin time assay". The vertical dotted lines indicate $1 \times N$, $2 \times N$, and $3 \times N$.

(16%) of mentions are unique. We observe that on average, unsupervised KRISS-Search has an edge over both PubMedBERT and BERT in terms of recall, precision, and F1. The error bars indicate $\pm$ 1 standard deviation. These error bars are large as the performance across CUIs varies.

As in Fig. 5 with the *same* evaluation type, Fig. 10 in A aggregates the results across 255 test query spans for the *related* evaluation type. Overall, it appears that the benefit of unsupervised KRISS-Search over PubMedBERT and BERT is still substantial when we make the evaluation less rigid and allow for more diverse spans.

In, Table 1 we compare the aggregate performance of unsupervised KRISS-Search and PubMedBERT. $\overline{\%\Delta}$ indicates that the average per-query percent improvement of unsupervised KRISS-Search over PubMedBERT is substantial. Furthermore, the win rates indicate that unsupervised KRISS-Search does better than PubMedBERT across most of the test queries. The P-values indicate that for number of nearest neighbors equals $1 \times N$, $2 \times N$, $3 \times N$, and both evaluation types, the benefit of unsupervised KRISS-Search over PubMedBERT is statistically significant.

Fig. 6 shows the precision-recall curves for the same performant prothrombin time assay example previously evaluated using the *same* evaluation type. We note that for this example, the benefit of unsupervised KRISS-Search (AUPRC = 0.60) over both PubMedBERT (AUPRC = 0.31) and BERT (AUPRC = 0.11) is substantial. Fig. 9 in A shows similarly beneficial results for the *related* evaluation type.

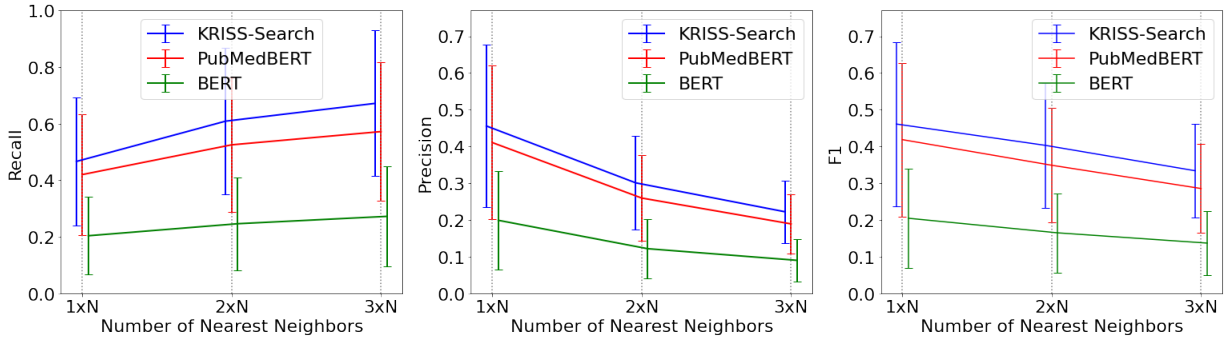


Figure 5: Mean recall (left), precision (center), and F1 (right) of unsupervised KRISS-Search (blue), PubMedBERT (red), and BERT (green) in retrieving concepts with the same CUI for number of nearest neighbors equals $1 \times N$, $2 \times N$, and $3 \times N$ across 255 test query spans. The error bars indicate ± 1 standard deviation. The three plots are staggered slightly to make the errors bars more visible.

Table 1: Comparison of unsupervised KRISS-Search and PubMedBERT with respect to recall across 255 test query spans. #NN refers to the number of nearest neighbors.

Eval Type	#NN	$\overline{\%\Delta}$	Win Rate	P-Val
Same	1xN	+ 24%	0.61	4.2e-3
	2xN	+ 29%	0.69	2.4e-5
	3xN	+ 31%	0.73	7.0e-7
Related	1xN	+ 26%	0.69	3.0e-4
	2xN	+ 35%	0.73	1.7e-6
	3xN	+ 35%	0.75	4.0e-7

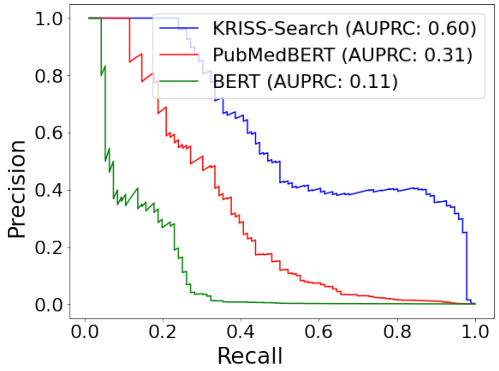


Figure 6: Precision-recall curves using the *same* evaluation type on an example query span with the text "PT", corresponding to the concept "prothrombin time assay".

Table 2: Average AUPRC scores from unsupervised KRISS-Search, PubMedBERT, and BERT across 255 test query spans. Results are presented as mean $\pm$ 1 standard deviation

Eval Type	Model	$\overline{\text{AUPRC}}$
Same	BERT	0.14 ± 0.12
	PubMedBERT	0.37 ± 0.23
	KRISS-Search	0.43 ± 0.25
Related	BERT	0.10 ± 0.09
	PubMedBERT	0.26 ± 0.19
	KRISS-Search	0.33 ± 0.23

Table 3: AUPRC comparison of unsupervised KRISS-Search and PubMedBERT.

Eval Type	$\overline{\%\Delta}$	Win Rate	P-Val
Same	+ 51%	0.71	4.5E-03
Related	+ 54%	0.76	1.6E-04

From Table 2 we observe that unsupervised KRISS-Search statistically significantly outperforms PubMedBERT ("P-Values" in Table 3). Although the average AUPRC decreases when moving from the *same* to the *related* evaluation type (as seen in Table 2), the average percentage change (as represented by $\overline{\%\Delta}$) increases (as seen in Table 3). This suggests that KRISS-Search's performance does not decline as steeply when transitioning from the *same* to the *related* evaluation type. We further assess whether this advantage persists when considering only unique mentions as positive spans. Utilizing the *same* evaluation type, we observe average AUPRCs of 0.24 ± 0.14, 0.20 ± 0.13, and 0.14 ± 0.11 for KRISS-Search, PubMedBERT, and BERT, respectively.

3.2 Supervised KRISS-Search

In concept retrieval on the n2c2 dataset, we outperform PubMedBERT and achieve an average F1 score of 0.68 ± 0.14 (using least confidence active learning, the sum of squares feature, and spatial refinement).

Table 5 shows an ablation study which demonstrates the utility of least confidence active learning (LC vs. the random sampling baseline), the sum of squares feature (SS), and spatial refinement (SpR). Furthermore, for the most performant configurations, the KRISS-Search embeddings outperform the PubMedBERT embeddings.

Fig. 7 shows the concept retrieval performance curves for an example "White Blood Count" span. This figure illustrates that as the supervised KRISS-Search iterations progress, incorporating human feedback consistently enhances the model's F1 performance. Furthermore, utilization of least confidence sampling (LC), sum of squares feature (SS) and spatial refinement (SpR) techniques results in less recall degradation while achieving the highest F1 score performance.

Table 4 shows that our method significantly outperforms BERT and also performs comparably to or outperforms PubMedBERT by an average of 5.6 F1 points. This is significant given that our method was not designed for NER. Our performance here indicates that supervised KRISS-Search can generalize to coarse-grained biomedical concepts and strict evaluation.

4 Conclusion

We demonstrate that unsupervised KRISS-Search outperforms existing embedding methods for biomedical interactive span recommendation. Supervised KRISS-Search utilizes humans-in-the-loop to achieve high levels of performance on both granular and course grain span recommendation. Future work will investigate whether KRISS-Search does indeed address the initial motivation - aiding programmatic data labeling as part of an interactive biomedical NLP system. Nonetheless, we envision KRISS-Search being broadly useful as a general purpose interactive biomedical search engine.

5 Limitations

One drawback of our method is that given a maximum span length, we always miss longer spans. For example, the BC2GM and JNLPBA NER datasets contain lengthy spans so we do not do as well on those tasks. Another drawback of our method is that it requires embedding the full corpus. One of our methods for making this tractable introduces another limitation - span filtering based on token types may discard spans that are useful to the user. Additionally, although we demonstrate that our method can be robust to training time (A.1), we have not explored principled methods for selecting the model checkpoint in supervised KRISS-Search, as the user does not label a validation set. Methods for making the process more rigorous should be explored, especially for out of distribution tasks.

6 Ethics Statement

The authors have evaluated the potential consequences of their research, including both positive and negative effects. Furthermore, the authors have ensured compliance with the guidelines outlined in the ACM Code of Ethics and Professional Conduct, and confirm that this work is in accordance with those principles.

References

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database-Issue):267–270.

Benedikt Boecking, Willie Neiswanger, Eric Xing, and Artur Dubrawski. 2021. Interactive weak supervision: Learning useful heuristics for data labeling. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Hunter Lang and Hoifung Poon. 2020. Self-supervised self-supervision by combining deep learning and probabilistic logic. *CoRR*, abs/2012.12474.

Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley, and Zhe Feng. 2021. Weakly supervised named entity tagging with learnable logical rules.

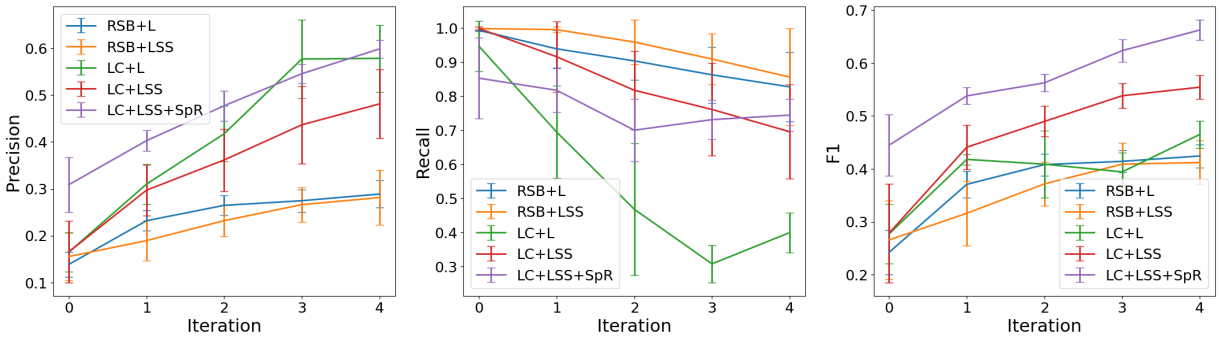


Figure 7: "White Blood Count" concept retrieval example across active learning iterations: precision (left), recall (middle), and F1 score (right). L stands for linear and denotes using the span embeddings without any additional features. LSS represents an additional sum of squares feature, which is the squared distance to the mean of positively labeled embeddings. LC denotes least confidence active learning, while RSB is the random sampling baseline. SpR represents spatial refinement human feedback. Values are mean ± 1 standard deviation across 3 active learning runs.

Table 4: Few-Shot biomedical NER results. For the KRISS-Search methods, the reported values represent the mean of 3 runs using different random seed example for active learning. PMB refers to PubMedBERT and KS refers to KRISS-Search. RSB, LC, and SpR refer to random sampling baseline, least confidence active learning, and spatial refinement respectively.

Dataset	BERT	PMB	KS (RSB)	KS (LC)	KS (SpR)
BC5-chem	0.69	0.73	0.67	0.82	0.84
BC5-disease	0.49	0.60	0.52	0.71	0.74
NCBI-disease	0.55	0.63	0.45	0.65	0.70
BC2GM	0.48	0.54	0.31	0.49	0.56
JNLPBA	0.55	0.59	0.37	0.47	0.53

Table 5: Ablation study with PubMedBERT and KRISS-Search on the concept retrieval task. The table compares least confidence active learning (LC) versus the random sampling baseline. It also measures how the sum of squares feature (SS), which denotes the squared distance to mean of the positively labeled embeddings, impacts performance. Furthermore, it measures the impact of spatial refinement human feedback (SpR). Values represent mean ± 1 standard deviation.

LC	SS	SpR	PubMedBERT	Kriss-Search
✓	✓	✓	0.66 ± 0.13	0.68 ± 0.14
✓	✓		0.61 ± 0.13	0.63 ± 0.13
✓		✓	0.57 ± 0.14	0.58 ± 0.14
✓			0.56 ± 0.14	0.56 ± 0.15
	✓	✓	0.34 ± 0.12	0.41 ± 0.16
	✓		0.33 ± 0.11	0.35 ± 0.11
		✓	0.33 ± 0.12	0.31 ± 0.12
			0.39 ± 0.12	0.37 ± 0.12

S. Liu, Wei Ma, R. Moore, V. Ganesan, and S. Nelson. 2005. Rxnorm: prescription for electronic drug information exchange. *IT Professional*, 7(5):17–23.

Yen-Fu Luo, Sam Henry, Yanshan Wang, Feichen Shen, Ozlem Uzuner, and Anna Rumshisky. 2020. The 2019 n2c2/UMass Lowell shared task on clinical concept normalization. *Journal of the American Medical Informatics Association*, 27(10):1529–e1.

Usman Naseem, Matloob Khushi, Shah Khalid Khan, Kamran Shaukat, and Mohammad Ali Moni. 2021. A comparative analysis of active learning for biomedical text mining. *Applied System Innovation*, 4(1).

Hoifung Poon, Hai Wang, and Hunter Lang. 2021. Combining probabilistic logic and deep learning for self-supervised learning. *CoRR*, abs/2107.12591.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. 2017a. Snorkel: Rapid training data creation with weak supervision. *CoRR*, abs/1711.10160.

Alexander J. Ratner, Stephen H. Bach, Henry R. Ehrenberg, and Chris Ré. 2017b. Snorkel: Fast training set generation for information extraction. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, page 1683–1686, New York, NY, USA. Association for Computing Machinery.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. 2020. A survey of deep active learning. *CoRR*, abs/2009.00236.

Kent A. Spackman, Ph. D, Keith E. Campbell, Ph. D, Roger A. Côté, and D. Sc. (hon. 1997. Snomed rt: A reference terminology for health care. In *J. of the American Medical Informatics Association*, pages 640–644.

Artur Yakimovich, Anaël Beaugnon, Yi Huang, and Elif Ozkirimli. 2021. Labels in a haystack: Approaches beyond supervised learning in biomedical applications. *Patterns*, 2(12):100383.

Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. 2022a. A survey on programmatic weak supervision.

Sheng Zhang, Hao Cheng, Shikhar Vashishth, Cliff Wong, Jinfeng Xiao, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022b. Knowledge-rich self-supervision for biomedical entity linking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 868–880, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xinyan Zhao, Haibo Ding, and Zhe Feng. 2021. Glara: Graph-based labeling rule augmentation for weakly supervised named entity recognition. *CoRR*, abs/2104.06230.

A Appendix

A.1 KRISS-Search Hyperparameters

For the conventional NER methods, we choose the following hyperparameters based on performance on the validation set. For KRISS-Search, the task of choosing hyperparameters for the single layer perceptron is less straightforward as we would like our method to generalize to settings where we do not have a labeled validation set for hyperparameter tuning. We hypothesize that we can include an L2 regularization term and train for many epochs without overfitting, eliminating the need for selecting a precise number of training iterations. We thus increase the default regularization coefficient from the scikit-learn MLP classifier default value of 1e-4 to 1e-3. Furthermore, we choose the Adam optimizer, hypothesizing that it is less sensitive than other optimization methods to initial learning rate. We selected an initial learning rate of 1e-4, a train batch size of 64, and 200 training iterations based on our hypothesis that these hyperparameters would result in training that is not sensitive to the number of training iterations. To validate this hypothesis, we also conducted additional experiments with only 100 training iterations, and found that the performance differences between the two sets of experiments were negligible. This suggests that our chosen hyperparameters are indeed robust and do not greatly affect the outcome of the training.

A.2 Recall of "PT" example using related evaluation type

Fig. 8 shows results for the same prothrombin time assay CUI example as was used in Fig. 4 but with the *related* evaluation type. We note here that the number of nearest neighbors corresponding to $1 \times N$, $2 \times N$, and $3 \times N$ is greater as expected.

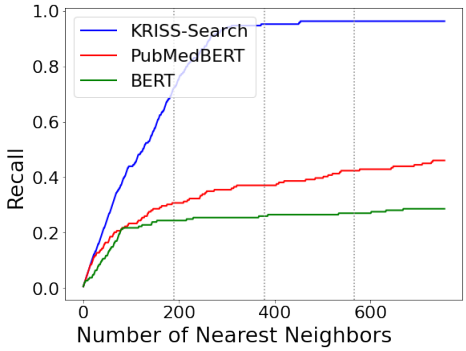


Figure 8: Recall using the *related* evaluation type on an example query span with the text "PT", corresponding to the concept "prothrombin time assay".

A.3 Aggregate recall, precision, and F1 using related evaluation type

Fig. 10 aggregates the results across 255 test query spans for the *related* evaluation type. The benefit of unsupervised KRISS-Search over PubMedBERT and BERT is substantial when we make the evaluation less rigid and allow for more diverse spans as compared to the *same* evaluation type.

A.4 AUPRC of "PT" example using related evaluation type

Fig. 9 shows the precision-recall curves for the prothrombin time assay example using the *related* evaluation type. As with the *same* evaluation type, the benefit of unsupervised KRISS-Search (AUPRC = 0.77) over both PubMedBERT (AUPRC = 0.26) and BERT (AUPRC = 0.13) is significant.

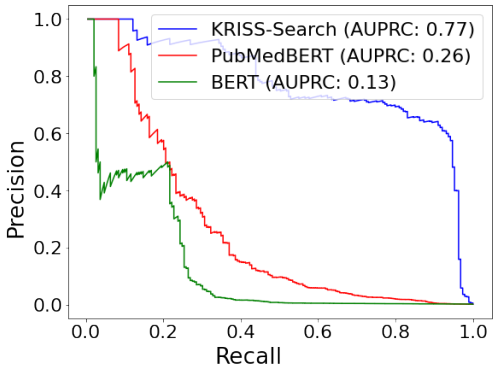


Figure 9: Precision-recall curves using the *related* evaluation type on an example query span with the text "PT", corresponding to the concept "prothrombin time assay".

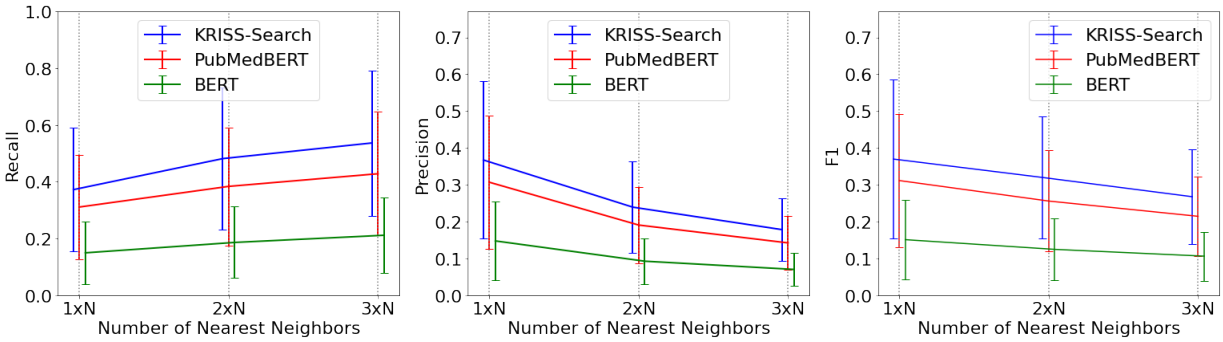


Figure 10: Aggregate recall (left), precision (center), and F1 (right) of unsupervised KRISS-Search (blue), PubMedBert (red), and BERT (green) in retrieving concepts with related CUIs for number of nearest neighbors equals $1 \times N$, $2 \times N$, and $3 \times N$ across 255 test query spans.