KRISS-Search: A Contextual Span Recommender for Biomedical Text

Louis Blankemeier[1] **Robert Tinn**[2] **Sid Kiblawi**[2] **Yu Gu**[2] **Akshay Chaudhari**[1]
Hoifung Poon[3] **Sheng Zhang**[3] **Mu Wei**[2] **Sam Preston**[2]
[1]Stanford University [2]Microsoft Health AI [3]Microsoft Research
{lblankem,akshaysc}@stanford.edu
{robert.tinn,sidkiblawi,aiden.gu,hoifung,
zhang.sheng,muhsin.wei,sam.preston}@microsoft.com

Abstract

Motivated by the scarcity of high-quality labeled biomedical text, as well as the success of data programming [12], where domain expert authored labeling functions provide weak labels for large datasets, we introduce *KRISS-Search*. We envision KRISS-Search increasing the efficiency of programmatic data labeling and providing broader utility as a general purpose interactive biomedical search engine. We first introduce *unsupervised KRISS-Search* and show that our method outperforms existing methods in recommending semantically similar spans (>50% AUPRC improvement relative to PubMedBERT [4]). We then introduce *supervised KRISS-Search* and, with simulated human interaction, demonstrate that we achieve high levels of performance on a task to classify spans as semantically similar or different, outperforming PubMedBERT by 2 F1 points. Finally, we demonstrate that our method performs competitively in low-resource biomedical NER.

1 Introduction

A critical challenge faced by practitioners of biomedical natural language processing is a paucity of high-quality labeled data. Manual annotation of biomedical text is a bottleneck for developers, since it requires far greater expertise than other domains. Techniques such as weak supervision [16, 2, 15, 9, 5] and active learning [8, 13] are promising methods to overcome this challenge. Programmatic data labeling [12, 11, 10], a source of weak supervision where domain experts develop heuristics (labeling functions) to provide noisy labels on large datasets, makes good use of domain expertise. However, developing such labeling functions that provide wide coverage often requires a comprehensive set of seed terms that serve as building blocks for these labeling functions. Augmenting the set of seed terms to increase the efficacy of the expert defined labeling functions can be time consuming and laborious.

Motivated by this challenge, we introduce *unsupervised KRISS-Search*. Reusing the KRISSBERT [17] embedding space, which is designed for biomedical entity-linking, unsupervised KRISS-Search addresses the following task - given a user-selected query span from a biomedical corpus, we return semantically similar spans to the user, where a span is a unique document identifier, start index, and end index positional 3-tuple. Next, we adapt unsupervised KRISS-Search to *supervised KRISS-Search* which leverages user feedback through active learning to refine the concept of similarity used for unsupervised KRISS-Search. 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 more flexible replacements for expert-developed labeling functions.

Our main contributions can be summarized as follows:

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

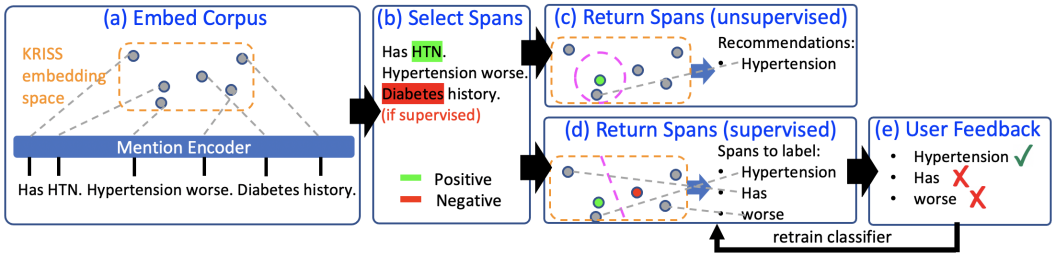


Figure 1: KRISS-Search method. (a): First, we embed the corpus. The KRISSBERT embedding space places mentions of the same concept (i.e. "HTN", "Hypertension") close and different concepts further apart. (b): The user selects spans to seed supervised and unsupervised KRISS-Search. For unsupervised KRISS-Search, the user selects a single positive query span. For supervised KRISS-Search, the user selects any number of positive and negative spans. (c) and (d) show how we use computations in the embedding space to return new spans. (c): In unsupervised KRISS-Search, we return nearest neighbors to the query span. (d): In supervised KRISS-Search, we use active learning to train a light-weight classifier to refine recommendations. This schematic shows examples closest to the decision boundary being returned for subsequent active learning. (e): A human provides feedback on the returned spans. We can then retrain the light-weight classifier and return to (d).

1. We demonstrate that unsupervised KRISS-Search outperforms PubMedBERT [4] 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 and by 54% in returning spans with similar associated CUIs (Table 2).
2. By extending unsupervised KRISS-Search to supervised KRISS-Search through user-feedback and active learning, we surface spans associated with specific concepts with F1 > 0.76, outperforming PubMedBERT by 2 points and BERT [3] by 13 points.
3. Despite our method not being specifically designed for named entity recognition (NER), we demonstrate that supervised KRISS-Search performs comparably to PubMedBERT on average in the low-resource biomedical NER setting.

2 Methods

In this paper, we focus our comparison of methods on training strategy and keep the BERT-base [3] architecture consistent across approaches. The methods reported in this paper can be summarized with the descriptors "contextual", "in-domain", "contrastive", and "interactive". *Contextual*: uses context to surface recommendations. *In-domain*: trained on in-domain data, in our case biomedical text. *Contrastive*: enforces similarity between semantically similar spans and dissimilarity between semantically different spans during training. *Interactive*: human interaction guides the generated recommendations. Each method - BERT, PubMedBERT, unsupervised KRISS-Search, supervised KRISS-Search - implements an additional descriptor in the order they were listed, with supervised KRISS-Seach implementing all four.

2.1 Unsupervised KRISS-Search

The unsupervised KRISS-Search task is as follows - given a user selected span from a biomedical corpus, which we refer to as the query span, we return related spans from the rest of the corpus. We modify KRISSBERT [17] to make generating spans for the full corpus computationally tractable as described in the Appendix A.1. Given a query span, we return a ranked list of related spans, ordered by the L2 distance of their embedding to the query span embedding. We reuse the KRISSBERT embedding space for our task as we hypothesize that the contrastive loss, as well as the domain specific pretraining, make the KRISSBERT embedding space particularly well suited for our task.

2.1.1 Evaluation

For evaluation of unsupervised KRISS-Search, we use the n2c2 dataset (2019 n2c2/UMass Lowell shared task 3) [7]. This dataset contains 100 discharge summaries labeled with CUI annotations.

We choose this dataset as it represents a domain shift from the PubMed abstracts that were used to train KRISSBERT. Additionally, n2c2 is annotated with diverse entities, including medical problems, treatments, and tests from established ontologies [6, 14].

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* in Tables 1 and 2) and (2) spans with associated CUIs that are closely related to the CUI associated with the query span (*related* in Table 1 and 2). Related CUIs are generated by sampling a parent CUI of the query-associated CUI and returning its children using the UMLS hierarchy [1]. The *same* experiments indicate how well each approach is at returning specific concepts of interest, while the *related* experiments measure how well each approach is at returning broader 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 and 2, as follows. For 255 CUIs with more than 25 mentions in the corpus and corresponding span embeddings, we randomly sample one span. We select CUIs that appear more than 25 times hypothesizing the difficulty of comparing approaches using low-prevalence CUIs. To evaluate the model performance, we compute average area under the precision-recall curve (AUPRC) values across the 255 test query spans for both the *same* and *related* experiments ($\overline{\text{AUPRC}}$ in Table 1). 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. We also compute average per-query percent AUPRC improvement of KRISS-Search compared to PubMedBERT ($\overline{\%\Delta}$ in Table 2), the frequency with which unsupervised KRISS-Search outperforms PubMedBERT with respect to AUPRC ("Wine Rate" in Table 2), 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-Value" in 2).

2.2 Supervised KRISS-Search

To incorporate user feedback, we train a light-weight classifier with KRISSBERT embeddings as input. We cache the KRISSBERT embeddings to reduce the latency that would results from fine-tuning KRISSBERT and embedding the corpus at each active learning iteration. Our pool-based active learning strategy is as follows. First, the user selects a small number of seed positive and negative 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.

2.2.1 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, simulating user feedback with the gold labels. We adopt a least confidence 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 append the L2 distance from the mean of the positively labeled embeddings to the KRISSBERT embeddings as an additional input feature. For these experiments, we use 28 concepts with greater than 100 mentions and corresponding embeddings. For evaluation, we compute performance using mentions that were not labeled during training.

2.2.2 Low-Resource Biomedical Named Entity Recognition

Although KRISS-Search is not designed for NER, we evaluate our method on the BLURB [4] NER datasets to ground our method in a well-understood task. Here, we adopt strict evaluation as is conventional in NER. We hypothesized that mean pooling aggregation does not sufficiently represent span boundaries and thus 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

Table 1: Average AUPRC scores from unsupervised KRISS-Search, PubMedBERT, and BERT across 255 test query spans.

Evaluation Type	Model	$\overline{\text{AUPRC}}$
Same	BERT	0.14 ± 0.129
	PubMedBERT	0.37 ± 0.233
	unsupervised KRISS-Search	0.43 ± 0.253
Related	BERT	0.10 ± 0.092
	PubMedBERT	0.26 ± 0.192
	unsupervised KRISS-Search	0.33 ± 0.233

Table 2: AUPRC comparison of unsupervised KRISS-Search and PubMedBERT.

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

and KRISS-Search, we equalize the number of labeled words used for training. We arbitrarily 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 the checkpoint with the best performance on the full validation sets. We forgo this approach with KRISS-Search, as we assume that the user has not labeled validation sets. We report results using the random sampling baseline (RSB in Table 3), as well as least confidence active learning (LC in Table 3).

3 Results

3.1 Unsupervised KRISS-Search

From Table 1 we observe that unsupervised KRISS-Search significantly outperforms PubMedBERT, and that this improvement is significant ("P-Value" in Table 2). We note that while average AUPRC decreases from the *same* to *related* evaluation type (Table 1), $\overline{\%\Delta}$ increases (Table 2).

3.2 Supervised KRISS-Search

In concept retrieval on the n2c2 dataset, we achieve the following average F1 score: 0.761 ± 0.204, which exceeds PubMedBERT by 2 points and BERT by 13 points. These results are notable as the number of negative spans outnumber the number of positive spans by more than 200x. The random sampling baseline and active learning without the distance feature perform worse.

Table 3 shows comparable performance of our method on average in low-resource NER. One drawback of our method in the strict evaluation context is that given a maximum span length, we always miss longer spans. BC2GM and JNLPBA contain lengthy spans so we do not do as well here. Nonetheless, on the other datasets, our method performs comparably or outperforms PubMedBERT. This is significant given that our method was not designed for NER. Our performance here indicates that supervised KRISS-Search can generalize to coarse-grain biomedical concepts and strict evaluation.

Table 3: Low-resource biomedical NER.

Dataset	BERT	PubMedBERT	KRISS-Search (RSB)	KRISS-Search (LC)
BC5-chem	69.85	73.25	68.39	83.02
BC5-disease	49.93	60.93	49.27	71.95
NCBI-disease	55.00	63.84	40.99	64.92
BC2GM	48.45	54.62	35.63	50.17
JNLPBA	55.30	59.74	36.31	48.48

4 Conclusion

We demonstrate that unsupervised KRISS-Search outperforms existing embedding methods for biomedical span similarity. Supervised KRISS-Search illustrates how with human interaction we can achieve high levels of performance on a task to retrieve similar spans and we can perform competitively in low-resource biomedical NER. 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.

References

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

[2] B. Boecking, W. Neiswanger, E. Xing, and A. Dubrawski. Interactive weak supervision: Learning useful heuristics for data labeling. In *International Conference on Learning Representations*, 2021.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Domain-specific language model pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779, 2020.

[5] H. Lang and H. Poon. Self-supervised self-supervision by combining deep learning and probabilistic logic. *CoRR*, abs/2012.12474, 2020.

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

[7] Y.-F. Luo, S. Henry, Y. Wang, F. Shen, O. Uzuner, and A. Rumshisky. The 2019 n2c2/UMass Lowell shared task on clinical concept normalization. *Journal of the American Medical Informatics Association*, 27(10):1529–e1, 09 2020.

[8] U. Naseem, M. Khushi, S. K. Khan, K. Shaukat, and M. A. Moni. A comparative analysis of active learning for biomedical text mining. *Applied System Innovation*, 4(1), 2021.

[9] H. Poon, H. Wang, and H. Lang. Combining probabilistic logic and deep learning for self-supervised learning. *CoRR*, abs/2107.12591, 2021.

[10] A. Ratner, S. H. Bach, H. R. Ehrenberg, J. A. Fries, S. Wu, and C. Ré. Snorkel: Rapid training data creation with weak supervision. *CoRR*, abs/1711.10160, 2017.

[11] A. J. Ratner, S. H. Bach, H. R. Ehrenberg, and C. Ré. 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, 2017. Association for Computing Machinery.

[12] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[13] P. Ren, Y. Xiao, X. Chang, P. Huang, Z. Li, X. Chen, and X. Wang. A survey of deep active learning. *CoRR*, abs/2009.00236, 2020.

[14] K. A. Spackman, P. D, K. E. Campbell, P. D, R. A. Côté, and D. S. (hon. Snomed rt: A reference terminology for health care. In *J. of the American Medical Informatics Association*, pages 640–644, 1997.

[15] A. Yakimovich, A. Beaugnon, Y. Huang, and E. Ozkirimli. Labels in a haystack: Approaches beyond supervised learning in biomedical applications. *Patterns*, 2(12):100383, 2021.

[16] J. Zhang, C.-Y. Hsieh, Y. Yu, C. Zhang, and A. Ratner. A survey on programmatic weak supervision, 2022.

[17] S. Zhang, H. Cheng, S. Vashishth, C. Wong, J. Xiao, X. Liu, T. Naumann, J. Gao, and H. Poon. Knowledge-rich self-supervised entity linking. *CoRR*, abs/2112.07887, 2021.

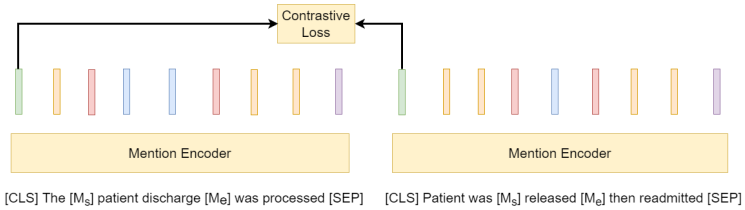
A Appendix

A.1 Efficiently Embedding the Corpus

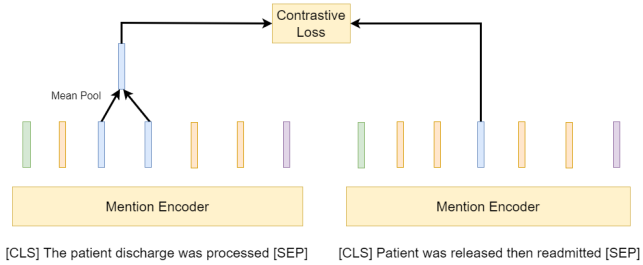
Unsupervised KRISS-Search and supervised KRISS-Search both require embedding the entire. However, in most cases, this is computationally intractable with KRISSBERT. KRISSBERT [17] uses the CLS token to represent a span's contextual embedding. To communicate the span of interest to the model, KRISSBERT places entity tokens between the span and its context. Therefore, generating embeddings for X spans requires X forward passes.

To overcome this limitation, we remove the entity flags from the mention representations. To produce span embeddings, we aggregate the final layer embeddings of tokens in the span. Fig. 2a shows how KRISSBERT uses entity tokens (corresponding embeddings shown in red) 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 are an example of a positive pair where "patient discharge" and "released" correspond to the same concept and are thus pushed together in the embedding space during contrastive training. The entity encoder is left unchanged. We train the entity encoder jointly with the mention encoder as is done in KRISSBERT as we hypothesize that the hierarchical UMLS information embedded in the entity encoder is useful for our task.

Our modifications greatly increase computation efficiency. If we pass 512 tokens through 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. 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 the selected hyperparameters optimize validation performance of the original model. Therefore, a slight loss of performance on the validation data is expected. As the goal of this paper is not entity linking, we leave re-selecting hyperparameters to future work.



(a) KRISSBERT mention encoder training with entity tokens. This is the approach used in the original paper.



(b) KRISSBERT mention encoder training without entity tokens. This is the strategy adopted for this paper.

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