TrainX – Named Entity Linking with Active Sampling and Bi-Encoders

Tom Oberhauser, Tim Bischoff, Karl Brendel, Maluna Menke,
Tobias Klatt, Amy Siu, Felix Alexander Gers, Alexander Löser
Beuth University of Applied Sciences
Berlin, Germany
{toberhauser, tkbischoff, karl.brendel, mmenke,
tobias.klatt, siu, gers, aloeser}@beuth-hochschule.de

Abstract

We demonstrate *TrainX*, a system for Named Entity Linking for medical experts. It combines state-of-the-art entity recognition and linking architectures, such as Flair and fine-tuned Bi-Encoders based on BERT, with an easy-to-use interface for healthcare professionals. We support medical experts in annotating training data by using active sampling strategies to forward informative samples to the annotator. We demonstrate that our model is capable of linking against large knowledge bases, such as UMLS (3.6 million entities), and supporting zero-shot cases, where the linker has never seen the entity before. Those zero-shot capabilities help to mitigate the problem of rare and expensive training data that is a common issue in the medical domain.

1 Introduction

Named Entity Linking is a well-studied task for decades (Ling et al., 2015). It includes recognizing and disambiguating mentions of entities in text against a catalogue or a knowledge base. However, training data is often missing and requires additional expensive labeling, especially in domains like medicine, where the availability of domain experts for rare diseases is limited. Moreover, novel as well as uncommon entities such as rare diseases might not have been part of the training data; in that case, the linker must solve a zero-shot scenario by disambiguating a mention never seen before. Existing easy-to-use annotation interfaces like prodigy[1] either fail to support entity linking annotations or have limited support for an end-user to find the correct entity in a large knowledge base. Further, they do not support the annotator by actively sampling relevant documents to save annotation time. Active-learning-enabled annotation tools, like INCEpTION (Klie et al., 2018), overcome this problem, but they are optimized for annotating multiple layers of linguistic features, which makes their user interfaces very complex and crowded. Using such tools leads to additional training costs for medical professionals.

Contribution We present *TrainX*, a system that consists of state-of-the-art entity recognition and linking architectures combined with an easy-to-use interface for healthcare professionals. Our system supports medical experts in annotating data and training models for medical named entity linking based on UMLS (Bodenreider, 2004) with more than 3.6 million entities. By using active sampling, we minimize labeling efforts. TrainX uses transfer learning by leveraging Bi-Encoders (Gillick et al., 2019; Wu et al., 2019; Logeswaran et al., 2019; Humeau et al., 2020) for disambiguation and a kNN-index to retrieve candidate entities within milliseconds. We mitigate issues caused by sparse training data by using zero-shot optimized techniques that can generalize beyond the labels seen in training. To our knowledge, this is the first named entity linking approach that combines an easy-to-use frontend with the transfer learning capabilities of recent BERT models. The system is licensed under Apache 2.0 and is available on GitHub[2].

[1] https://prodi.gy/

[2] https://github.com/DATEXIS/TrainX

Proceedings of the 27th International Conference on Computational Linguistics, pages 64–69
Barcelona, Spain (Online), December 12, 2020.

2 Demonstrating Medical Named Entity Linking

In this section, we demonstrate the usage of TrainX for a medical entity linking scenario. Although we show a domain-specific task, TrainX is not limited to the medical domain but can be easily adapted to any entity linking use case with a knowledge base that contains names and short descriptions for every entity. Figure 1 shows the usage of TrainX in an example scenario where existing GOLD annotations are available. First, the annotator begins a new session or resumes an existing one (1). In the case of a new session, she uploads a new dataset (2). After the upload, she obtains sampled documents from the dataset, in order to annotate them (3). The samples view (4) allows her to add new USER annotations (green) or to view/edit GOLD annotations provided in the dataset (yellow). By clicking on an annotation, she can examine details of the linked entity and make a correction if needed by using the annotation helper (5); the modified samples are henceforth USER annotations and marked in green. A full-text search on the UMLS supports the annotator to interactively explore the knowledge base in order to speed up the annotation/examination/correction workflow. By clicking on the checkmark (6), she can mark the entire sample as correct, or she can use the arrows above to request as many further samples as she likes. When one round of annotation is finished, she uploads the annotated samples (7) and starts the training phase (8), while the system will apply the model to the newly adjusted data. She can query the training status at any time (9). When training is finished, she retrieves the newly processed samples (10) and is returned to the samples view, where the predictions of the newly trained model are shown as PRED annotations in blue (11). Now, she can further correct and/or add annotations and iterate the process. A video of this demonstration is available under `https://youtu.be/XAt94UNEEQ4`.

Figure 1: Usage of the TrainX system to train and evaluate the entity linker on a mixture of given GOLD and added USER labels. The screenshots show the menu, where a user requests samples or starts the training and the sample view where she can edit annotations.

3 Named Entity Linking with Active Sampling and Bi-Encoders

A system component overview is shown in Figure 2. Prior to training, the system needs to be initialized with the knowledge base (UMLS in our case) and an optional set of pre-training documents (1). After the initialization, the user can upload her documents (2) and annotate (3) them using the support of the annotation helper (4). The user is supported by an active sampling of further samples to annotate and correct. Next, the updated annotations and the current model are sent to training component where

the named entity recognizer and linker are now (re-)trained on the supplied data (5). After the training succeeded, the newly trained model is used to recognize and link mentions in the uploaded documents to provide feedback to the user (6).

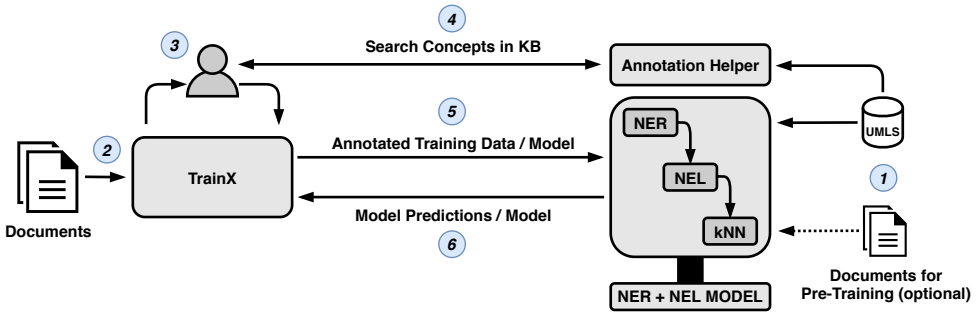

Figure 2: Workflow of the TrainX system.

Named Entity Recognition and Linking with Bi-Encoders The recognition step is the first step of an entity-linking pipeline. A high recall is crucial because the linker will not be able to disambiguate mentions that have not been found by the recognizer in the first place. We chose the Flair-framework (Akbik et al., 2018) because it has proven to achieve state-of-the-art results (Devlin et al., 2018). We implemented a Bi-Encoder based on the work by Wu et al. (2019) and Humeau et al. (2020). The Bi-Encoder uses fine-tuned BERT models to project mentions and entities in a common dense vector space to allow retrieval based on vector similarity. The projection is enforced using a cross-entropy loss function on both encoders' output that rewards a high similarity between the mention and the matching entity representation. In contrast to other entity linking architectures such as the entity linker from the "ScispaCy" framework (Neumann et al., 2019), the Bi-Encoder approach is fully focused on solving zero-shot problems (Wu et al., 2019). Also, it allows us to apply confidence measurements needed for our active sampling mechanisms. We chose not to implement an additional cross-encoder as proposed by Wu et al. (2019), Logeswaran et al. (2019) and Humeau et al. (2020) due to computational efficiency (Kurz et al., 2020). The Bi-Encoder consists a mention encoder $y_m = (pool(T_1(m))$ and an entity encoder $y_e = (pool(T_2(e))$. T_1 and T_2 are BERT models, and m and e are sequences of WordPiece tokens that encode mention and entity, respectively. The pooling method $pool()$ aggregates the resulting tensor into single vector representations y_m and y_e by using the vector of the `[CLS]` token as a representation of the whole token sequence.

Training the Bi-Encoder Our system works with three types of labels: GOLD for labels from (optional) pre-training data, USER labels that have been created or updated by the user, and PRED labels that were predicted from the entity linker. We train on GOLD and USER labels. For the mention encoder T_1, the mention and its context are encoded by using two special tokens to mark the beginning and end of a mention i.e. `[CLS] context_left [MS] mention [ME] context_right [SEP]`. The input of the entity encoder T_2 is the name of the entity, followed by a textual description, i.e. `[CLS] name [ENT] description [SEP]`. The description for every entity is generated by concatenation of all English descriptions within the UMLS for that concept, starting with the longest one. The maximum number of WordPiece tokens is a hyper-parameter of the model (Wu et al., 2019). Following Humeau et al. (2020), we fine-tune all BERT layers except the embeddings to minimize the cross-entropy loss for a vector of the logits $y_{m_i} \cdot y_{e_1}, \ldots, y_{m_i} \cdot y_{e_i}, \ldots, y_{m_i} \cdot y_{e_n}$ for every (m_i, e_i) in the batch B where $|B| = n$.

Candidate Retrieval For a given mention embedding, the system retrieves an entity by performing a kNN-search based on a normalized dot product between that embedding and all of the concept embeddings. Because an exact kNN search will be too slow in practice for large amounts of data, we will not only examine retrieval performance based on exact kNN, but also on an approximate kNN approach, namely "Hierarchical Navigable Small World graphs" (HNSW) (Malkov and Yashunin, 2020) using the

implementation of Facebook AI Similarity Search (Faiss) (Johnson et al., 2017). HNSW outperforms other approximate kNN approaches in terms of quality/speed trade-off (Aumüller et al., 2017).

Hyperparameters Our Bi-Encoder fine-tunes BERT-base models with a learning rate of 5-e5, as suggested by Devlin et al. (2018). As further hyperparameters, we chose a maximum input length of 50 WordPiece tokens for both encoders, a batch size of 128 samples and 100 learning-rate warmup steps. The HNSW indexes are initialized with $m = 16$, $efConstruction = 100$ and $efSearch = 100$.

Active Sampling The goal of our human-in-the-loop process is two-fold: First, the user should become familiar with the quality of the model and second, the system should support the user to improve the performance of the model quickly. The system applies the model to the data after the training and enables the user to approve or correct results. Thereby it selects samples based on the confidence of the named entity recognizer and named entity linker. For each annotation, we calculate the confidence $conf_{ann}$ by aggregating the confidence of the NER $conf_{NER}(ann)$, as provided by the Flair framework, and the confidence of the entity linker based on margin sampling (Scheffer et al., 2001) by taking into account the difference between the retrieved candidate entities at first and second places (e_{ann_1}, e_{ann_2}) with respect to the query vector $q = T_1(ann)$: $conf_{ann} = conf_{NER}(ann) + (cos(q, e_{ann_1}) - cos(q, e_{ann_2}))$. The documents are sampled based on their least confident annotations.

4 Evaluation and Discussion

Evaluation on MedMentions We chose the publicly available "MedMentions-ST21pv" dataset (Mohan and Li, 2019) containing 4,392 annotated abstracts of PubMed articles with 203,282 annotations of 25,419 unique concepts. We use the pre-defined train/test/dev split. The test set contains annotations for 3,590 concepts that have not been in the training set which allows us to evaluate zero-shot capabilities. As named entity recognizer, we selected the Flair framework (Akbik et al., 2018) and trained it on BIOES tagging until the early-stop mechanism of Flair stopped the training process. The scores of this model resulted in a precision of 69.2, a recall of 69.0 and an F1 of 69.1. In Table 1, we report the retrieval scores of the isolated entity linking component in comparison to a BM25-based Elasticsearch[3] full-text index baseline. Table 1 also provides an end-to-end evaluation of the whole pipeline using the micro-averaged A2W weak annotation match metric proposed by Cornolti et al. (2013).

	UMLS concepts ∈ MedMentions					full English UMLS				
	full test set			only zero-shot		full test set			only zero-shot	
	BM25	exact	HNSW	exact	HNSW	BM25	exact	HNSW	exact	HNSW
R@1	40.9	63.2	63.2	50.3	50.0	21.7	44.4	43.4	26.5	25.3
R@20	73.7	86.2	86.1	76.3	76.0	50.7	73.6	71.7	55.6	53.1
End-to-End (NER + NEL)										
Precision	-	47.4	47.4	-	-	-	32.1	31.1	-	-
Recall	-	45.7	45.7	-	-	-	30.9	30.0	-	-
F1	-	46.5	46.5	-	-	-	31.5	30.6	-	-

Table 1: Retrieval Performance – The upper half shows the the recall@k performance of the entity linker compared to the BM25 baseline for exact kNN and HNSW indexes of all the UMLS concepts that can be found within the MedMentions dataset (25,419) or the full English UMLS (3.6 Million). A separate zero-shot evaluation shows the performance of the linker on concepts that it has not seen during training. The lower half provides an end-to-end evaluation of the whole pipeline.

Discussion The Bi-Encoder outperforms the BM25 based approach by a margin of more than 20 percentage points. With respect to the size of the concept database of 3.6 million concepts and given that UMLS still contains many ambiguities (Shooshan et al., 2009), the Bi-Encoder is still able to link to the correct concept 26.5 percent of the times even though it has never seen it during training (zero-shot). On the full test set, which contains a mixture of seen data and zero-shot, the Bi-Encoder was able to link to

[3]https://www.elastic.co/

the exact entity in 44.4% of all cases. HNSW reduces the retrieval performance about one percentage point at worse but speeds up the query process to 3ms instead of 600ms per query. The inclusion of the named entity recognizer reduces the recall to 30.9% and results in an overall F1 score 31.5% for exact kNN and 30.6% for HNSW. The zero-shot performance indicates that the underlying Bi-Encoder is able to generalize beyond concepts seen in training to mitigate problems caused by sparse training data. Therefore, our further work will focus on the optimization of the Bi-Encoder and the named entity recognition step in order to better adapt to sparse-training data situations.

Acknowledgements

Our work is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) under grant agreement 01MK20008D (Service-Meister).

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2017. ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms. In *International Conference on Similarity Search and Applications*, pages 34–49. Springer.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. 2013. A Framework for Benchmarking Entity-Annotation Systems. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 249–260, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning Dense Representations for Entity Retrieval. *arXiv preprint arXiv:1909.10506*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *8th International Conference on Learning Representations, ICLR*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734*.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In Dongyan Zhao, editor, *COLING 2018, The 27th International Conference on Computational Linguistics: System Demonstrations, Santa Fe, New Mexico, August 20-26, 2018*, pages 5–9. Association for Computational Linguistics.

Nadja Kurz, Felix Hamann, and Adrian Ulges. 2020. Neural Entity Linking on Technical Service Tickets. *arXiv preprint arXiv:2005.07604*.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design Challenges for Entity Linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-Shot Entity Linking by Reading Entity Descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460.

Yu A. Malkov and D. A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, Apr.

Sunil Mohan and Donghui Li. 2019. MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts. In *Automated Knowledge Base Construction (AKBC)*.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August. Association for Computational Linguistics.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active Hidden Markov Models for Information Extraction. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, IDA '01, page 309–318, Berlin, Heidelberg. Springer-Verlag.

Sonya E Shooshan, James G Mork, and A Aronson. 2009. Ambiguity in the UMLS Metathesaurus. In *Tech rep, US National Library of Medicine*.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot Entity Linking with Dense Entity Retrieval. *arXiv preprint arXiv:1911.03814*.