Cross-Lingual UMLS Named Entity Linking using UMLS Dictionary Fine-Tuning

Anonymous ACL submission

Abstract

We study cross-lingual UMLS named entity linking, where mentions in a given source language are mapped to UMLS concepts, most of which are labeled in English. We propose a general solution that can be easily adapted to any source language and demonstrate the method on Hebrew documents. Our cross-lingual framework includes an offline unsupervised construction of a bilingual UMLS dictionary and a per-document pipeline which identifies UMLS candidate mentions and uses a fine-tuned pretrained transformer language model to filter candidates according to context.

Our method exploits a small dataset of manually annotated UMLS mentions in the source language and uses this supervised data in two ways: to extend the unsupervised UMLS dictionary and to fine-tune the contextual filtering of candidate mentions in full documents. Our method addresses cross-lingual UMLS NEL in a low resource setting, where the ontology is large, there is a lack of descriptive text defining most entities, and labeled data can only cover a small portion of the ontology. We demonstrate results of our approach on both Hebrew and English. We achieve new state-of-the-art results on the Hebrew Camoni corpus, +8.9 F1 on average across three communities in the dataset. We also achieve new SOTA on the English dataset MedMentions with +7.3 F1.

1 Introduction

Public health practices are becoming increasingly digital, with tools to explore scientific sources of information such as medical literature and online health communities rising in popularity. Such tools are essential in offering insights to researchers, providing information to patients and to their caregivers. Reliable identification of mentions of biomedical concepts in free text is a key technique to enable robust mining of such textual resources. Named-Entity Recognition (NER) is the task of classifying entities in text to high level classes (Person, Organization, Gene, Disease, Treatment, etc.). Named-Entity Linking (NEL) seeks to additionally classify entity mentions in text into specific concepts according to an existing reference list or knowledge base. We focus in this work on biomedical NEL, i.e., identifying mentions referring to biomedical concepts such as disorders and drugs and linking them to normalized concepts, for example, those listed in the Unified Medical Language System (UMLS) ontology. Biomedical NEL has been mostly studied in English. Other languages present additional challenges because terms in the ontology are described in English. We address cross-lingual NEL (xNEL) which consists of mapping mentions in a source language to concepts labeled and described in a different target language. We focus on UMLS xNEL, where mentions in the source language (we specifically test Hebrew, see Appendix A for a Hebrew tagging example) are mapped to UMLS concepts. We aim for a general solution that can be adapted to any source language. We operate in a low resource setting, where the ontology is large, text describing most entities is not available, and labeled data can only cover a small portion of the ontology. We also consider different genres of text to be annotated, ranging from consumer health medical articles in popular web sites to scientific biomedical articles.

Our main contributions are: (1) We provide a general framework for cross-lingual UMLS NEL that can be adapted to source languages with few pre-requisites; our method includes four steps (a) offline unsupervised learning of a language-specific UMLS dictionary; for each document: (b) generation of candidate mentions, (c) high-recall matching of candidate mentions to UMLS concepts and (d) contextual relevance filtering of (candidate, concept) pairs. Steps (c) and (d) take advantage of multi-lingual pre-trained transformer language models (PLMs). (2) Our method exploits a small annotated corpus of documents in the source language and genre annotated manually for UMLS mentions (a few thousands annotated mentions). This training data is split to support (a) the extension of the unsupervised UMLS dictionary with corpus-salient entity names and (b) fine-tune the contextual ranking and filtering of (candidate mentions, concept) pairs. We find that the step of UMLS dictionary fine-tuning boosts NEL performance and identify a clear tradeoff in allocating training data between lexicon extension and contextual fine-tuning; (3) We demonstrate results of our approach on both Hebrew and English. We achieve new SOTA on the Hebrew Camoni corpus (Bitton et al., 2020) with +8.87 F1 and on the English dataset MedMentions (Mohan and Li, 2019) with +7.3 F1[1].

2 Previous Work

Biomedical NEL is challenging because the underlying ontology (most often UMLS) is extremely large and the acquisition of annotated training data requires rare and expensive expertise. Loureiro and Jorge (2020) presented MedLinker, a tool for improving biomedical NEL by predicting the semantic type of a medical concept mention and filtering out candidates of the wrong type. MedLinker was tested on the MedMentions task of concept linking (Mohan and Li, 2019), improving above TaggerOne (Leaman and Lu, 2016), the baseline model for MedMentions which did not use deep learning. MedLinker splits the end to end task of entity linking into two stages - candidate recognition and linking. For candidate matching, it combines a BiLSTM-CRF model for contextual matching with an approximate dictionary matching method to increase recall. In the cross-lingual setting, dictionary matching is not applicable. We report our results on the same MedMentions dataset in 5.2.

Past work has shown that using in-domain text can provide additional gains over general-domain language models (Gu et al., 2020). Therefore, recent work (BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019)) addressed biomedical NEL, focusing on pre-training models on scientific/medical text. Liu et al. (2021) developed SapBERT, a pre-training scheme which exploits the graph structure of the UMLS ontology and aims at learning an encoding of medical mentions that can align with synonym relations in the UMLS graph. Combining the SapBERT objective with pre-training on biomedical text of PubMedBERT (Gu et al., 2020) boosts results on NEL. Experimental results demonstrated that SapBERT outperforms many domain-specific BERT-based variants (BioBERT and SciBERT) on the BC5CDR dataset. Although our model focuses on cross-lingual NEL, it also applies to English documents. We compare our results to these approaches on BC5CDR and MedMentions (Tables 4 and 3).

Indexing of the abundant biomedical scientific literature requires precise detection of medical concepts. Mohan et al. (2021) developed a low-resource recognition and linking model of biomedical concepts called *LRR* aimed at generalizing to entities unseen at training time, and incorporating linking predictions into the mention segmentation decisions. This BERT-based model achieved SOTA results on the MedMentions task. In our work, we adopt the *LRR* bottom-up candidate generation approach (see 4.2). We address the main drawback of the approach by incorporating a UMLS dictionary fine-tuning technique which extends the list of candidate pairs (source expression, CUI) on a portion of the training data. We elaborate on the motivation for the technique in 4.5 and demonstrate its contribution in ablation experiments (see 5.4).

xNEL, the problem of grounding mentions of entities in a source language text into a different target language knowledge base (typically English), has been addressed in recent years, with a range of promising techniques. When the source and target languages operate over different alphabets and sound systems, both translation and transliteration of terms (which is a noisy process even when done by people) must be handled. Bitton et al. (2020) curated the Camoni corpus, an annotated resource of Hebrew posts from online health communities (OHCs), where noisy text (as opposed to scientific text) introduces additional challenges. Many user queries mention medical terms, which are very likely to include noisy transliterations. For example, the Hebrew query equivalent to “How do I know I have fibromyalgia?” does not return any results in the search engine of the Camoni online community when ‘fibromialgia’ is transliterated. Bitton et al. (2020) introduced MDTEL (Medical Deep Transliteration Entity Linking) for Hebrew-English NEL on noisy text in OHCs, and tested it on the Camoni corpus. MDTEL adopts a four-step approach - consisting of an offline unsuper-

[1]Our code is publicly available https://github.com/

vised Hebrew UMLS dictionary learning, candidate mention generation, high-recall matching and filtering of matching mentions. We adopt MDTEL's unsupervised UMLS dictionary matching, which uses an attention-based recurrent neural network encoder-decoder that maps UMLS from English to Hebrew (either a Hebrew translation or transliteration of the concept). We introduce new methods for candidate generation, high-recall matching and contextual relevance filtering, relying on multilingual pre-trained language model (mBERT). Our new components lead to significant performance improvement over MDTEL on the Camoni corpus.

3 Task Formulation

Given input language L and target language L_t, a database of medical concepts $C_{L_t} : L_t^* \to CUI$ is a function from concept names in L_t to concept IDs (CUIs). Using C_{L_t}, we want to learn a function F from a span in input language L and its context to a CUI. We identify a translated dictionary, $C_L : L^* \to CUI$. C_L is the "translated" version of the medical concepts database C_{L_t}. We learn C_L by mapping the medical terms in L_t to terms in L. Given mapping C_L, we aim to learn:

$$F : L^* * L^* \to CUI \cup \{\perp\}$$

where $\perp$ is a special code denoting a non-medical term. F differs from C_L as it addresses the variability and ambiguity of the task by depending on the context as well as the span. Given text $W = (w_1, ..., w_n)$, where $w_i \in L$, for every span $s_{i,j} = (w_i, ..., w_j) \subseteq W$, we would like to compute $F(W, s_{i,j})$, where $0 \leq j - i < k$ (we limit the span sizes to at most k), that is, we want to predict the concept associated with a span within a text in L. Provided a dataset A_L exposing a subset of F combined with linguistic knowledge and generalization capabilities of neural models, we aim at learning a larger portion of function F.

4 Model Architecture

Our end-to-end xNEL model (Fig.1) consists of four consecutive stages: (1) **multilingual UMLS mapping**: generate UMLS dictionary C_L (see 4.1) based on the method of Bitton et al. (2020); (2) **candidate generation**: consider all spans of up to k words as candidate mentions and compute vector representations for both mentions and concepts (see 4.2); (3) **high recall matching**: use a semantic similarity based score function to generate the top matching entities with high recall (see 4.3) and (4) **contextual relevance modeling**: encode each candidate into a context-dependent vector representation using a pre-trained transformer-based language model fine tuning process (see 4.4).

Our approach attempts to avoid three types of mistakes: (1) **morphological and transliteration noise**, where candidate terms in the source language might be extracted due to a transliteration or morphological error and matched with UMLS entities, (2) **contextual errors**, where candidate terms which are not medical terms when considering the context might be matched with UMLS entities, and (3) **partial UMLS tagging**, where candidate terms which are not full medical terms in the text but rather more general UMLS mentions might be tagged as UMLS concepts (e.g., in the mention "flu vaccine", "flu" should not be tagged). The first challenge is addressed by learning a high-recall C_L dictionary with generalization capabilities, trained both on translation and transliteration data; the second, is addressed by an mBERT-based contextual language model; the third, by systematic consideration of all spans up to size k as candidates as part of the candidate generation and contextual relevance components.

4.1 Multilingual UMLS Mapping

The first step of our model is offline, fully unsupervised, and based on the method of (Bitton et al., 2020): we generate a mapping C_L between medical concept names in source language L to their corresponding CUIs. An attention-based character-based recurrent neural network encoder-decoder is used to create a list of *⟨UMLS term in English, term in language L⟩* so that each UMLS term in English is matched with both transliterated and translated forms in L. This is done without the need of manually annotated data and results in a noisy mapping C_L of source language medical terms and their CUIs.

4.2 Candidate Generation

Given a document in L where we want to identify UMLS mentions, the candidate generation step begins with pre-processing: we normalize the source text documents from annotated data A_L and the target UMLS concepts from C_L by transforming all string values to lower case and removing delimiters. We then generate a list of overlapping candidate mention spans, ranging in length according to the max length parameter k (*i.e.,* $1, ..., k$. See

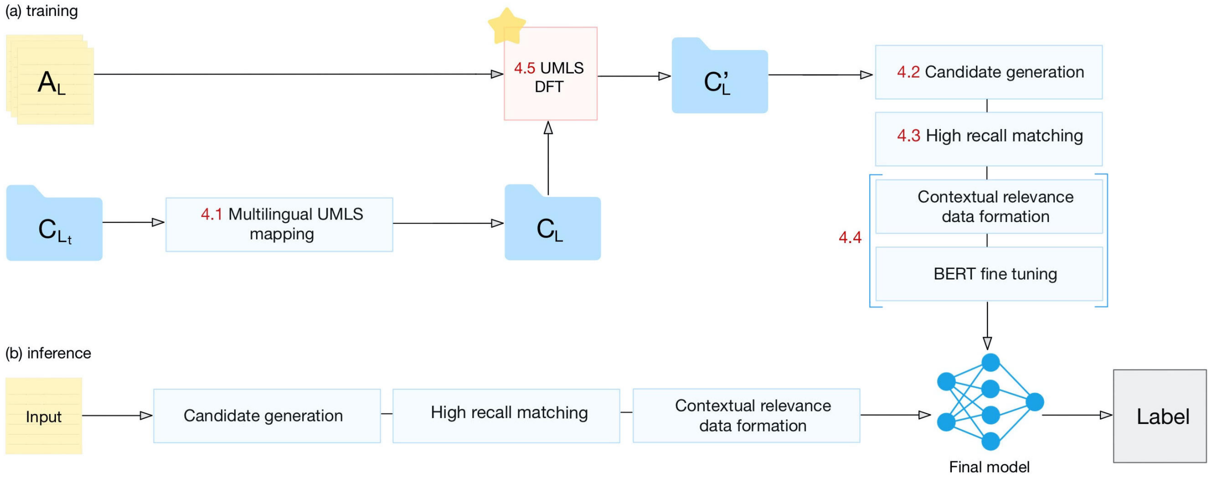

Figure 1: End-to-end pipeline overview. Training process is depicted in section (a), inference process is depicted in section (b).

Appendix B for details). We exclude spans starting or ending with stop words. We then represent both the spans and the concepts as tf-idf character n-gram (1 to 3-gram) vectors using sklearn's implementation (Pedregosa et al., 2011). Empirical experiments showed that tf-idf encoding improved recall in candidate generation compared to bag of words encoding (see Appendix C for a comparison between the two representations using both Hebrew and English datasets).

4.3 High Recall Matching

The high recall matcher (HRM) receives the vector representations from the candidate generator and computes a similarity score between each span and all concept names in C_L using cosine similarity (see Appendix C for comparison against Manhattan score function). We then select the top m matches per span with score over a threshold th (see Appendix D for hyper-parameters). This results in a high recall list of candidate matches.

4.4 Contextual Relevance Modeling

At this step, we want to predict which spans returned from the high recall matcher are true biomedical concepts. We use multilingual BERT (m-BERT) (Jacob Devlin, 2019), a 12 layer transformer that was trained on the Wikipedia pages of 104 languages (including Hebrew) with a shared word piece vocabulary. M-BERT does not use any marker denoting the input language, and does not include explicit mechanism to encourage translation equivalent pairs to have similar representations. We fine-tune m-BERT on a binary classification task on our training data: each candidate mention span returned from the HRM is centered in its context from the original doc, *i.e.,* W_s words to the right of the span and W_s words to the left of the span, creating a window surrounding the candidate mention. The classifier takes as input the window, the HRM's decision on which concept is represented by the mention in the window, and the true verdict of whether the candidate mention is indeed an occurrence of the concept. We utilize m-BERT's QA format as follows: the question (medical concept c) and the reference text (window w) are packed into the input, and provide the binary label as answer of whether or not c is a medical mention in context w: $[CLS]\ w\ [SEP]\ c\ [SEP]$. This fine-tuning step consists of adding an additional output layer on top of the pre-trained m-BERT model to adapt it to the biomedical NEL task.

4.5 UMLS Dictionary Fine-Tuning

We introduce a UMLS dictionary fine-tuning technique where some of the data in A_L is removed from the training dataset and used to directly expand the learned dictionary C_L. We reserve $R\%$ of the training data A_L to fine-tune C_L generating C'_L (see Fig. 1): from this chunk of A_L, we add each mention in the tagged data as new pairs (mention in L, CUI).

For example, suppose our training data consists of 10 tagged documents and our UMLS dictionary C_L contains 100 concepts. Given $R = 10\%$, our UMLS dictionary fine-tuning technique will require one tagged document d (10% of the 10 docs in the training set) to be used for fine-tuning C_L.

We go over every tagged pair (m, c) from doc d, where m is a mention in doc d and c is the UMLS concept the annotators tagged m. If $m \notin C_L$, we add m to C_L with the CUI of c. Suppose doc d contained 15 such tags, we will obtain an augmented C'_L containing $100 + 15 = 115$ concepts.

We cannot use this portion of data for later training of our model, since after fine-tuning we are guaranteed to get a perfect match for all the spans in the documents used for fine-tuning (thus creating bias of the HRM).

Although this process decreases the overall size of the input dataset for contextual relevance fine-tuning, it improves the recall of the HRM and adds more positive examples for the BERT training process. We elaborate more on this trade-off in 5.4.2. This approach allows us to improve recall on synonyms and abbreviations that were not originally in our UMLS dictionary, with genre-specific terminology observed in the training data (as evident from the experiment shown in Table 6).

5 Experiments

We test our approach both on cross-lingual UMLS Linking using the Camoni dataset of Hebrew consumer health data and on English UMLS Linking using MedMentions and BC5CDR, which include scientific papers in the bio-medical field.

5.1 Camoni Corpus

The Camoni corpus was curated by Bitton et al. (2020) for the analysis of the MDTEL system. Camoni is an Israeli social network in Hebrew aimed at patients with chronic diseases and their family members (Camoni). Camoni serves about 20,000 registered members and 100,000 unique visitors per month. The digital platform is organized into 39 disease-specific communities. Bitton et al. (2020) extracted text from three communities (diabetes, sclerosis, and depression), for a total of 55,000 posts and 2.5 million tokens, and constructed an annotated dataset in which 1,000 mentions of UMLS terms were annotated. Bitton et al. (2020) proposed a high recall matcher based on a fuzzy string matching algorithm introduced in prior work to perform the matching between the spans and medical entities. Table 1 compares our HRM results (recall) with MDTEL for each community (diabetes, depression, sclerosis).

We observe that our candidate generation method (adopting the LRR bottom-up approach and mBERT similarity matching) significantly improves the recall of the HRM (from about 74% in MDTEL to about 82% overall). We believe that the use of the tf-idf character n-gram vectorization before applying the cosine similarity function as means of comparison helped us achieve better results compared to MDTEL's method which only applied the cosine similarity.

Model	Community	Recall %
MDTEL	Diabetes	76.6
Our model	Diabetes	**82.0**
MDTEL	Depression	74.1
Our model	Depression	**83.5**
MDTEL	Sclerosis	70.0
Our model	Sclerosis	**81.0**

Table 1: High recall matcher performance on Camoni corpus.

In the end to end linking task, our model achieves much higher precision (98% vs. 77%) at the cost of slightly lower accuracy but much improved F-score 84 vs 74. Table 2 compares the performance of MDTEL with our model on the end to end linking task for each community.

5.2 MedMentions

MedMentions (Mohan and Li, 2019) is a corpus of Biomedical papers annotated with mentions of UMLS entities. The corpus consists of 4,392 papers (Titles and Abstracts) randomly selected from papers released on PubMed in 2016, that were in the biomedical field, published in the English language, and had both a Title and an Abstract available. MedMentions contains over 350,000 linked mentions, annotated by a team of professional annotators with rich experience in biomedical content curation. We focus on MedMentions ST21pv (21 Semantic Types and Preferred Vocabularies), a subset of the full annotations containing 203,282 mentions and restricting the concepts to a 2.3M large subset of the full ontology (UMLS ST21pv). Each concept in this subset is associated with one of 21 selected semantic types, or to one of their descendants in the semantic type hierarchy.

We compare our performance to other models' results on MedMentions ST21pv in Table 3. We improve on the latest SOTA LRR (Mohan et al., 2021), achieving +7.3 F1.

Our recall was similar to LRR, however our model achieved highly improved precision, 76.4

Model	Community	Accuracy %	Precision %	Recall %	F1 %
MDTEL	Diabetes	**97.0**	71.0	**75.0**	73.0
Our model	Diabetes	89.2	**98.3**	73.8	**84.3**
MDTEL	Depression	**99.0**	77.0	73.0	75.0
Our model	Depression	90.8	**97.7**	**76.9**	**86.0**
MDTEL	Sclerosis	**98.0**	82.0	**71.0**	76.0
Our model	Sclerosis	86.3	**98.3**	67.8	**80.3**

Table 2: Intrinsic evaluation performance of our model on Camoni corpus.

Model	Accuracy %	Precision %	Recall %	F1 %
TaggerOne	-	47.1	43.6	45.3
MedLinker	-	48.4	50.1	49.2
LRR	-	63.0	52.0	57.0
Our model	74.8	**76.4**	**55.5**	**64.3**

Table 3: Performance of different models on the MedMentions dataset. "-": not reported in the paper.

compared to 63. We believe this improvement can be attributed to our UMLS dictionary fine-tuning technique, which provides an extended list of candidates and thus more examples for the mBERT fine-tuning process for contextual relevance. Mohan et al. (2021) mention the need to improve recall for cases where the mentions are indirect or too abbreviated to generate a good lexical match from the entity knowledge base, which is exactly what our technique helps improve. For example, our process picked up in the training data that the abbreviation *mrn* is tagged as *messenger rna* (CUI C0035696), which was not originally present in the UMLS dictionary for English.

5.3 BC5CDR

The BC5CDR corpus (Li et al., 2016) consists of 1,500 PubMed articles with 4,409 annotated chemicals, 5,818 diseases and 3,116 chemical-disease interactions. Each entity annotation includes both the mention text spans and normalized concept identifiers, using MeSH (Lipscomb, 2000) as the controlled vocabulary (MeSH is part of the UMLS ontology). Compared to MedMentions which contains annotations of general medical concepts, BC5CDR is topic-specific, containing only annotations of chemicals and diseases. BC5CDR is also much smaller, consisting of just 1,500 articles compared to the 4,392 annotated papers of MedMentions. BC5CDR has a total of 13,343 linked mentions compared to 203,282 in MedMentions ST21pv.

We compare our model's performance to other models using BC5CDR's test set in Table 4, while

Model	Dataset	F1 %
BioBERT	BC5CDR	88.6
SciBERT	BC5CDR	90.0
SapBERT	BC5CDR-d	**93.5**
Our model	BC5CDR	73.0

Table 4: Performance of different models on the NER task using BC5CDR dataset.

Accuracy %	Prec. %	Recall %	F1 %
81.6	88.4	62.2	73.0

Table 5: Performance of our model on BC5CDR dataset.

Table 5 details our full results (additional evaluation metrics).

We observe that domain-specific pre-trained transformers help improve results on BC5CDR (93.5 F-measure vs. 73.0 for our model). The subset of semantic types covered in this dataset is much more technical (chemicals and chemical-disease interactions) than those covered in MedMentions, even though both BC5CDR and MedMentions include documents in the same genre of scientific biomedical articles. This difference is evidenced in the ablation study presented below. It explains why specialized language models trained on the biomedical domain lead to much improved performance compared to our model which uses the general mBERT. We hypothesize that using SapBERT combined with our model could enhance performance on this dataset and leave this for future work.

5.4 UMLS Dictionary Fine-Tuning Ablation Study

In this section, we test several factors impacting the contribution of UMLS dictionary fine-tuning to our tagger's performance. First, we test the technique on two different datasets and evaluate its benefits depending on the dataset size. Next, we test a range of UMLS dictionary fine-tuning percentage values (R) and discuss the trade-off between this value and the end to end performance of our linker.

5.4.1 Dataset Size Impact

We tested the UMLS dictionary fine-tuning technique on English datasets MedMentions and BC5CDR across 5 random seeds and found that it improved recall on both, but impacting MedMentions much more than BC5CDR due to a much smaller number of added concepts in BC5CDR, 209 compared to 3,294 in MedMentions (see Table 6). The difference in the number of added concepts could be explained by the fact that BC5CDR is much smaller, thus the decrease in training data size counteracts the small number of concepts being added to the UMLS dictionary. To test this hypothesis, we took a subset of MedMentions of the same size as BC5CDR (annotation-wise: 8,575 in total), see Table 7 for results averaged across 5 random seeds. The results suggest that the size of the dataset directly affects the number of concepts added to our UMLS dictionary (227 added in the MedMentions subset, very close to the 209 added in BC5CDR), which in turn impacts the HRM's recall: the improvement in recall is very similar between the two datasets, +1.37 for BC5CDR, +1.7 for MedMentions subset.

5.4.2 The Recall-Accuracy Tradeoff

We first observe that our UMLS dictionary fine-tuning technique can only improve the high recall matching performance (Section 4.3) since an annotation that we do not have a good semantic match for from UMLS will be a missed match without UMLS DFT. Similarly, an annotation for which we do have a good semantic match will be found regardless of whether we utilize UMLS DFT or not. Thus, UMLS dictionary fine-tuning helps us find non-semantically similar matches that we would have otherwise missed, meaning that the higher R is - the higher the recall of the HRM should be. However, there is a trade-off between the recall gained from the annotations utilized for UMLS dictionary fine-tuning and the overall performance

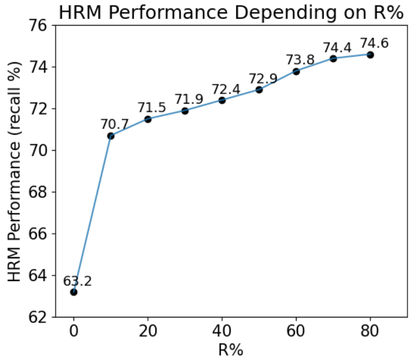

Figure 2: HRM Performance (recall%) on MedMentions dataset depending on the value of R.

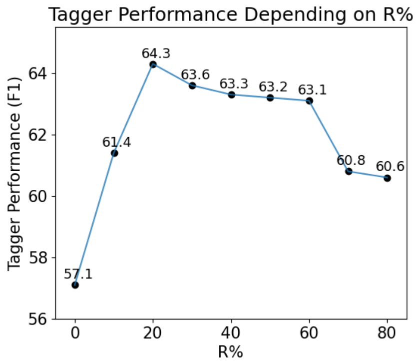

Figure 3: Tagger Performance (F1) on MedMentions dataset depending on the value of R.

of the linker, since the annotations used for fine-tuning are examples that the contextual model will be missing during fine-tuning. We explore this trade-off and compare the performance of the high recall matching component with the final tagging results of our model using different values of R on the MedMentions dataset. Figure 2 shows that there is a clear trend of increased recall of the HRM as R increases. However, Figure 3 shows the complexity of the trade-off since the tagger's performance reaches a peak and then begins to drop as R increases. The contextual model fine-tuning improvement plateaus after a certain amount of training examples, demonstrating the benefit of multi-task adaptation of pre-trained models which converge rapidly. The data efficiency of the contextual relevance fine-tuning process allows the UMLS dictionary fine-tuning technique to help improve end to end linking results.

6 Conclusion

In this work we explored the task of cross lingual named entity linking in the biomedical field. We

Dataset	UMLS DFT	Added Concepts	Recall %
MedMentions	✗	0	63.2
MedMentions	✓	3,294	**71.5**
BC5CDR	✗	0	74.13
BC5CDR	✓	209	**75.5**

Table 6: Number of added concepts per dataset and the average performance of the HRM with and without UMLS dictionary fine-tuning, across 5 random seeds. "✗": UMLS DFT not used, "✓": UMLS DFT used.

Dataset	UMLS DFT	Added Concepts	Recall %
MedMentions subset	✗	0	62.7
MedMentions subset	✓	227	**64.4**

Table 7: We took a subset of MedMentions the same size as BC5CDR (8,575 annotations). We report the number of added concepts and the average performance of the HRM with and without UMLS DFT across 5 random seeds. "✗": UMLS DFT not used, "✓": UMLS DFT used.

describe a pipeline to detect and link mentions of UMLS concepts in documents in Hebrew or in English, which improves upon existing methods. The key characteristics of our approach are (1) it distinguishes candidate generation from linking; (2) it uses the sophisticated unsupervised UMLS dictionary construction using the character-level RNN model introduced in Bitton et al. (2020) which takes into account both translation and transliteration but extends this dictionary with a portion of the training data mentions; empirical analysis of this dictionary augmentation method demonstrates its importance in end to end linking performance; (3) it adopts the bottom-up systematic generation of candidates from Mohan et al. (2021) and improves it by using a compact tf*idf ranking of the candidates (char n-gram) which helps reduce memory allocation; (4) it uses a multi-lingual pre-trained language model (mBERT) to fine-tune a contextual relevance model to filter a list of high-recall candidate matches. Our framework for cross-lingual UMLS NEL can easily be adapted to any source language and does not rely on any descriptive text for the entities.

We compared our performance to baseline approaches on the Camoni dataset in Hebrew (Bitton et al., 2020), and the MedMentions (Mohan and Li, 2019) and BC5CDR English datasets. Our end-to-end approach achieves SOTA results on Camoni in Hebrew and MedMentions in English with significant improvements. For BC5CDR, we observe that the small size of the dataset prevents our dictionary augmentation technique from reaching its potential and models trained on specialized biomedical text (PubMedBert with SapBert training objective) obtain better coverage. Such specialized training is, however, not available in a multi-lingual setting.

For future work, we intend to test whether utilizing language-specific BERT models instead of multilingual BERT (*e.g.,* swapping m-BERT with the recently released AlephBERT (Seker et al., 2021), a Hebrew version of BERT) could improve results on the Hebrew Camoni corpus. In addition, taking into account the SapBERT objective which exploits the UMLS graph structure as part of either fine-tuning or pre-training in Hebrew could lead to improved generalization capabilities. Finally, exploring datasets with additional source languages will help understand the capabilities of our multilingual pipeline. The CLEF eHealth challenges (Névéol et al., 2017, 2018) are good candidates for such analysis.

References

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*.

Yonatan Bitton, Raphael Cohen, Tamar Schifter, Eitan Bachmat, Michael Elhadad, and Noémie Elhadad. 2020. Cross-lingual Unified Medical Language System entity linking in online health communities. *Journal of the American Medical Informatics Association*, 27(10):1585–1592.

Camoni. The camoni global network.

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. *arXiv preprint arXiv:2007.15779*.

Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. https://github.com/google-research/bert/blob/master/multilingual.md.

Robert Leaman and Zhiyong Lu. 2016. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics*, 32(18):2839–2846.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Carolyn E Lipscomb. 2000. Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online. Association for Computational Linguistics.

Daniel Loureiro and Alípio Mário Jorge. 2020. Medlinker: Medical entity linking with neural representations and dictionary matching. In *European Conference on Information Retrieval*, pages 230–237. Springer.

Sunil Mohan, Rico Angell, Nick Monath, and Andrew McCallum. 2021. Low resource recognition and linking of biomedical concepts from a large ontology. *arXiv preprint arXiv:2101.10587*.

Sunil Mohan and Donghui Li. 2019. Medmentions: a large biomedical corpus annotated with umls concepts. *arXiv preprint arXiv:1902.09476*.

Aurélie Névéol, Aude Robert, Robert Anderson, Kevin Bretonnel Cohen, Cyril Grouin, Thomas Lavergne, Grégoire Rey, Claire Rondet, and Pierre Zweigenbaum. 2017. Clef ehealth 2017 multilingual information extraction task overview: Icd10 coding of death certificates in english and french. In *CLEF (Working Notes)*.

Aurélie Névéol, Aude Robert, Francesco Grippo, Claire Morgand, Chiara Orsi, Laszlo Pelikan, Lionel Ramadier, Grégoire Rey, and Pierre Zweigenbaum. 2018. Clef ehealth 2018 multilingual information extraction task overview: Icd10 coding of death certificates in french, hungarian and italian. In *CLEF (Working Notes)*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Shaked Greenfeld, and Reut Tsarfaty. 2021. Alephbert: A hebrew large pre-trained language model to start-off your hebrew NLP application with. *CoRR*, abs/2104.04052.

A Hebrew UMLS Tagging Example

Figure 4 illustrates the process of linking Hebrew data to UMLS concepts. The given post was taken from the Camoni sclerosis community and translates to:

"Hello, recently, my gait has deteriorated and I was suggested to begin Botox treatment to release the muscles and prevent spasticity. Has anyone here undergone such treatment? Does it help? is there a risk that such a treatment will greatly weaken the muscle, causing the exact opposite action?".

The 6 spans (colored) are linked to to 4 different CUIs of Unified Medical Language System medical concepts.

B Span Length Selection (k)

Span length represents the number of words we select from the input text and may or may not represent a medical concept (UMLS). This definition is used in the candidate generation step (see Section 4.2), where we create representations of all possible spans in the text and match them to top ranking concepts.

In order to define the max span length parameter k of the model, we performed a simple analysis of the annotated span lengths per dataset. As can be seen in Figures 5, 6 and 7, the most common length values tagged are generally 1 or 2. Taking into account computational limitations of using large span lengths, we chose $k = 3$. Note that even if the maximal span length selected is smaller than the maximal medical term length in the target dataset C_L, it is still possible to match source spans to such medical terms since our scoring function does not exclude matches based on length comparison (see Section 4.3).

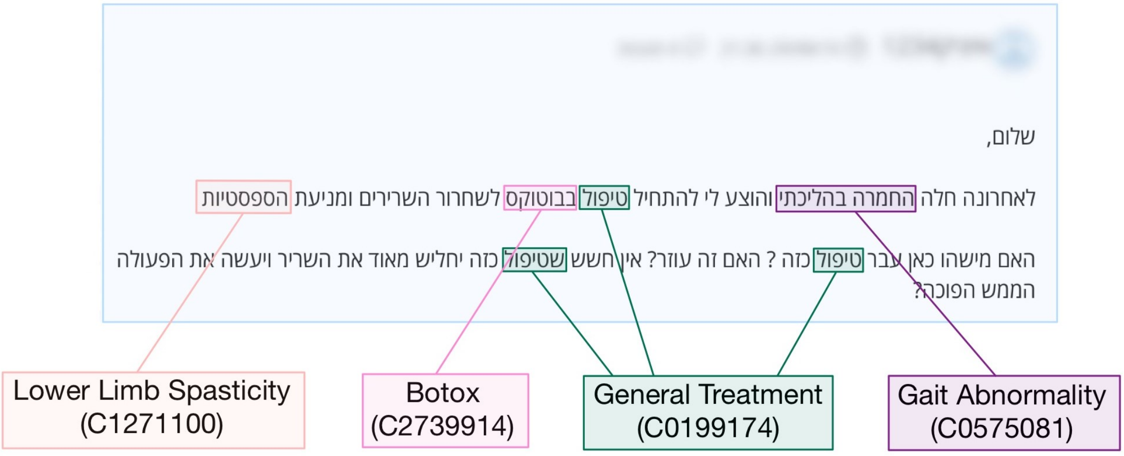

Figure 4: A forum post from the Camoni sclerosis community. The post contains 37 words, and 6 spans that link to 4 different CUIs of Unified Medical Language System medical concepts. Notice that a span can consist of more than 1 word (like the term matched to "gait abnormality") and a single CUI can be referenced from several places in the same post.

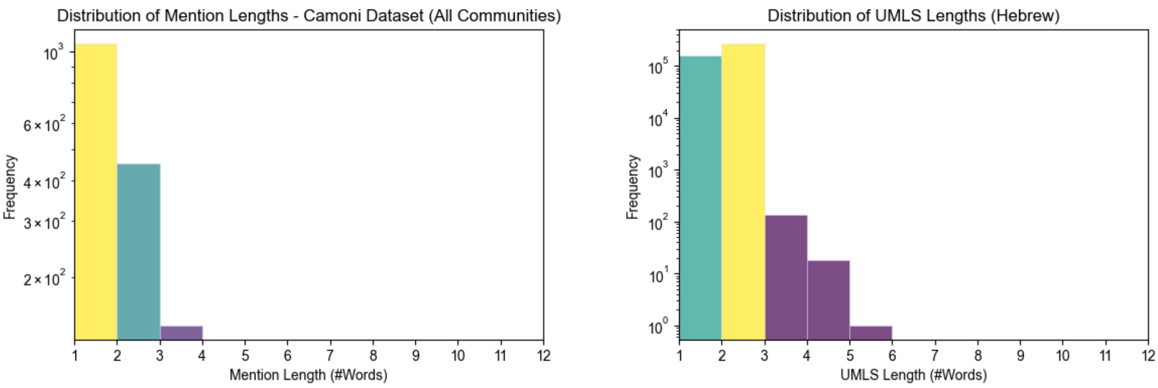

Figure 5: Distribution of Camoni Mention and UMLS Lengths (in words)

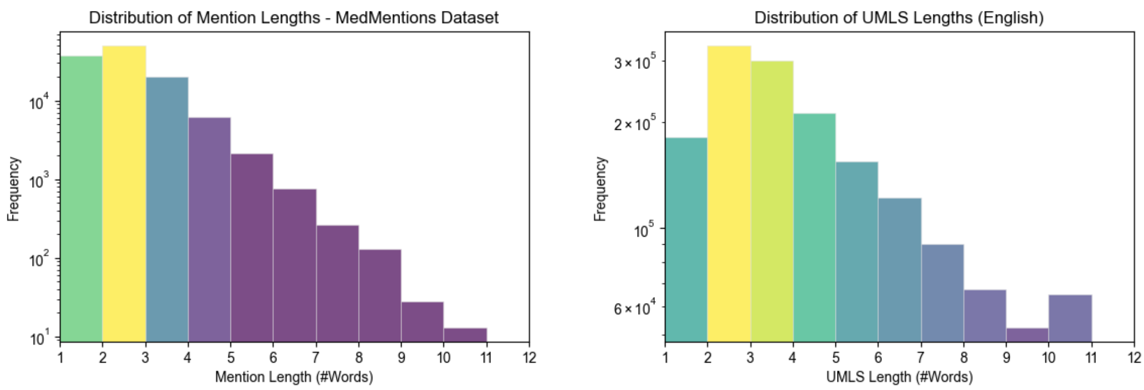

Figure 6: Distribution of MedMentions Mention and UMLS Lengths (in words)

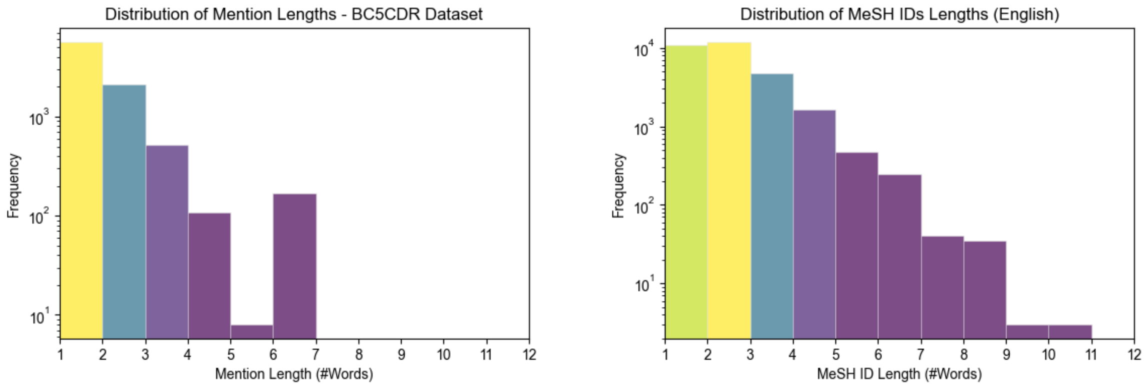

Figure 7: Distribution of BC5CDR Mention and MeSH ID Lengths (in words)

Vectorizer	Score Function	Recall %
Tf	Cosine	69.3
Tf	Manhattan	68.4
Tf-Idf	Cosine	**70.7**
Tf-Idf	Manhattan	69.7

Table 8: Performance of the HRM using two different vectorization methods and two different score functions on MedMentions dataset.

C Vectorization and Score Function Methods Comparison

We compared the performance (recall %) using two different score functions: (1) cosine similarity and (2) Manhattan distance, and two different vectorization techniques: (1) term frequency (tf) and (2) tf-idf (term frequency * inverse document frequency). We used character unigram, bigram and trigram analysis in all the reported cases (Table 8).

We hypothesize that the improvement stems from Idf penalizing frequent words by taking the log of {number of docs in the corpus divided by the number of docs in which the term appears}, where in our context, a 'doc' is either a span of text or a UMLS concept from C_L. Since no stop words can appear at either the start or end of the span/concept, we increase the odds of having meaningful words comprising each 'doc'. The tf-idf method may contribute to this further because it not only focuses on the frequency of words present in the corpus (tf, bag of word) but also provides an importance weight to them.

Vectorizer	Score Function	Recall %
Tf	Cosine	81.5
Tf	Manhattan	81.8
Tf-Idf	Cosine	**82.0**
Tf-Idf	Manhattan	81.9

Table 9: Performance of the HRM using two different vectorization methods and two different score functions on Camoni dataset (diabetes community).

D Hyper-Parameters

Table 10 describes all the hyper parameters' values we used in our model's implementation.

HP	Description	Value
m	top matches parameter of the high recall matcher (Section 4.3)	50
th	threshold of selecting possible matched concepts for the spans (Section 4.3)	0.4
W_s	window size per side of the candidate mention (Section 4.4)	2
R	UMLS dictionary fine-tuning percentage (Section 4.5)	20
-	the model's learning rate	$2e-5$
-	train epochs	3
-	batch size	32

Table 10: Hyper parameters (HPs) used in our model's implementation.