Biomedical Entity Representation with Graph-Augmented Multi-Objective Transformer

Andrey Sakhovskiy$^{1,2,3}$, Natalia Semenova$^{1,5}$, Artur Kadurin$^5$, Elena Tutubalina$^{1,2,4,5}$

$^1$Sber AI, $^2$Kazan Federal University, $^3$Skoltech
$^4$ISP RAS Research Center for Trusted Artificial Intelligence, $^5$AIRI

Correspondence: {andrey.sakhovskiy,tutubalinaev}@gmail.com

Abstract

Modern biomedical concept representations are mostly trained on synonymous concept names from a biomedical knowledge base, ignoring the inter-concept interactions and a concept’s local neighborhood in a knowledge base graph. In this paper, we introduce Biomedical Entity Representation with a Graph-Augmented Multi-Objective Transformer (BERGAMOT), which adopts the power of pre-trained language models (LMs) and graph neural networks to capture both inter-concept and intra-concept interactions from the multilingual UMLS graph.

To obtain fine-grained graph representations, we introduce two additional graph-based objectives: (i) a node-level contrastive objective and (ii) the Deep Graph Infomax (DGI) loss, which maximizes the mutual information between a local subgraph and a high-level graph summary. We apply contrastive loss on textual and graph representations to make them less sensitive to surface forms and enable intermodal knowledge exchange. BERGAMOT achieves state-of-the-art results in zero-shot entity linking without task-specific supervision on 4 of 5 languages of the Mantra corpus and on 8 of 10 languages of the XL-BEL benchmark.

1 Introduction

Biomedical concepts, such as diseases, symptoms, drugs, genes, and proteins, are critical for many biomedical applications, including drug discovery (Wu et al., 2018; Khrabrov et al., 2022; Zitnik et al., 2018), clinical decision making (Sutton et al., 2020; Peiffer-Smadja et al., 2020), and biomedical research (Lee et al., 2016; Tutubalina et al., 2017; Fiorini et al., 2018; Soni and Roberts, 2021; Sakhovskiy et al., 2021; Sakhovskiy and Tutubalina, 2022). The same biomedical concept may have multiple nonstandard names, abbreviations, and misspellings. Medical concept normalization (MCN), also known as medical concept linking, is a task of mapping entity mentions to a large set of medical concept names and their unique identifiers (CUIs) from a knowledge base (KB). The biomedical domain is characterized by extensive KBs such as the Unified Medical Language System (UMLS) (Bodenreider, 2004), which includes over 166 lexicon/thesauri with over 4M concepts and 15M concept names in 27 languages.

The development of meaningful and robust biomedical entity representations continues to be a challenging task for language models (LMs). Recent studies have probed LMs trained on biomedical texts in English and discovered that domain-specific pre-trained language models (PLMs), such as BioBERT (Lee et al., 2020) and SciBERT (Beltagy et al., 2019), exhibit high levels of bias and lack synonym awareness (Sung et al., 2021). For Spanish, the second language by number of concept names in UMLS, PLMs pre-trained on clinical data fall short compared to the simplistic sparse baseline in the MCN task (Alekseev et al., 2022).

Textual triples from a KB are commonly used to incorporate knowledge into neural networks with metric learning and contrastive learning framework (Phan et al., 2019; Miftahutdinov et al., 2021; Liu et al., 2021a; Yuan et al., 2022; Zhou et al., 2022). Positive and negative pairs are created using head and tail terms of the same or different concepts, as illustrated in Fig. 1, where maux de tête is a French synonym of headache but differs from sharp headache (headache is a broader concept for sharp headache). Graph-based representations are another way to represent biomedical knowledge with concepts as nodes and relationships as edges. Inspired by semantic matching methods like TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015), Yuan et al. (2022) proposed a method to integrate term-relation-term similarities into backbone LM. However, this approach doesn’t fully utilize inter-concept interactions since it learns from individual relation triplets rather than performing an aggregation over the whole con-
In this paper, we present Biomedical Entity Representation with Graph-Augmented Multi-Objective Transformer (BERGAMOT) which utilizes PLMs and graph neural networks (GNNs) to capture inter-concept and intra-concept interactions from the multilingual UMLS graph. As shown in Fig. 1, the BERGAMOT architecture includes four losses: (i) a textual term-term contrastive loss that learns from positive and negative concept name pairs; (ii) a node-node contrastive loss that learns on concept nodes to distinguish between nodes based on their local subgraphs in larger KG; (iii) DGI loss that lets a GNN distinguish between factually accurate (present in the knowledge graph) concept subgraphs and corrupted ones; (iv) an intermodal contrastive loss that enables mutual information exchange between textual and graph encoders. The source code and pre-trained models are freely available at: https://github.com/Andoree/BERGAMOT.

2 Related Work

MCN and entity representations There are several conventional approaches to address an MCN problem. The most common is the entity classification (Niu et al., 2019; Lou et al., 2020) into a small number of target concepts. However, UMLS and similar KBs may include millions of different concepts arranged in a hierarchical structure. Another popular approach is ranking mentions or concepts by the mutual similarity term learnt from positive and negative pairs on some corpora (Mondal et al., 2019). Aside from MCN methods, a plethora of works are focused on features and representations of entity mentions based on syntax, morphology, and synonyms (Aronson, 2001; Van Mullegien et al., 2016; Dermouche et al., 2016). Mondal et al. (2019) chose a straightforward convolutional layer as an encoder. The network was trained with the triplets of a disease mention, as well as positive and negative concept candidates. The supervised BioSyn model (Sung et al., 2020) maximizes the likelihood of synonym appearance among the most similar 20 terms. Morphology was encoded with a character-level TF-IDF representation to obtain a sparse similarity score. The distance between BioBERT (Lee et al., 2020) CLS tokens was utilized as a high-level dense similarity. The final similarity score is a weighted sum of both sparse and dense similarities. DILBERT (Miftahutdinov et al., 2021) introduces novel negative sampling strategies for a triplet loss, utilizing the hierarchical structure of the UMLS. Both BioSyn and DILBERT are limited to a single language (English) and a small concept subset within a specific terminology. Entity representation learning may be augmented with external knowledge from domain-specific KBs (Phan et al., 2019; Michalopoulos et al., 2021; Liu et al., 2021a,b; Yuan et al., 2022). (Phan et al., 2019) proposes encoding contextual meaning, conceptual meaning, and the similarity between synonyms during the representation learn-
Graph neural networks in biomedical domain

GNNs have gained attention in the last decade, with comprehensive surveys exploring their applications in various fields, including biology and medicine. GNNs are successfully applied to a wide range of graph- and node-level tasks in fields of drug discovery and material design (Wu et al., 2018; Khraibov et al., 2022; Zitnik et al., 2018), medicine (Ahmed-Aristizabal et al., 2021; Gligorijević et al., 2021), and Question-Answering (QA) tasks related to knowledge graphs (KG) (Vollmers et al., 2021; Chen et al., 2020). Graph-level biomedical tasks are provided in Open Graph Benchmark (Hu et al., 2020), while KG-related evaluation is not that straightforward. The majority of approaches focused on QA evaluation or conventional KG link prediction. We can divide the considered methods into three groups: LM- or KG-based and joint LM+KG. Despite approximately similar training data (subsets of UMLS) and backbone (SciBERT, BART (Lewis et al., 2020)), the approaches are entirely different due to data preparation procedures and finetuning. Chang et al. (2020) proposed a completely KG-based biomedical benchmark. They trained TransE, CompLEx (Trouillon et al., 2016), DistMult, SimplE (Kazemi and Poole, 2018), and RotatE (Sun et al., 2019) on the SNOMED-CT dataset to compare the results with static Snomed2Vec (Agarwal et al., 2019) and Cui2Vec (Beam et al., 2020) baselines. While static methods did not surpass any KG-trained methods, KG-based models performed remarkably worse in comparison with LM-based ones due to the lack of a text encoder. An alternative approach to benchmarking is provided in the LM+KG architectures’ evaluation. QA-GNN (Yasunaga et al., 2021) and GreaseLM (Zhang et al., 2022) achieved state-of-the-art scores on MedQA (Jin et al., 2021). However, they are both trained in a manner of fine-tuning LM with KG-augmentation. This strategy does not allow us to completely adapt models to the MCN task.

3 Background and architecture

3.1 UMLS knowledge graph

Let $V$ denote a set of all concepts present in a knowledge base and $R$ denote a set of possible relation types between concepts from $V$. Knowledge graphs, such as UMLS, usually store relational information in the form of relation triplets $(h, r, t) \in V \times R \times V$. Let $E$ denote a set of all unique relation triplets from a given KG. Thus, the UMLS graph can be defined as an oriented edge-labeled graph $G = G(V, E, R)$ with a set of nodes $V$, a set of labeled oriented edges $E$, and a set of possible edge labels $R$. For each concept $c \in V$, UMLS presents a set of $k$ synonymous concept names $S_c = \{s_{c1}, s_{c2}, \ldots, s_{ck}\}$. For each name from $S_c$, UMLS stores the label of the language it came from.

3.2 Self-alignment pretraining

A reasonable and straightforward way to learn an informative representation space of biomedical entities is to represent textual knowledge from KG in the form of positive and negative term pairs and optimize some contrastive learning loss function.

In this work, we adopt the self-alignment pretraining (SAP) procedure proposed by Liu et al. (2021a). To enrich the training procedure with harder negative samples, SAP employs online hard mining for valid triplets (Mikolov et al., 2013; Gillick et al., 2019). During SAP, the model is encouraged to produce similar representations for all terms that represent the same concept (share the same CUI). At each pretraining step, we sample a set $T$ that consists of $N$ positive samples $(c, s_{ci}, s_{cj}) \in V \times S_c \times S_c$. Given $T$, SAP constructs all possible term triplets $(s^p, s^n, s^n)$ such that $p = a$ and $n \neq a$. $s^p$ is called an anchor term; $s^p$ is a positive term for $s^n$ (i.e., $s^p$ and $s^n$ are synonymous terms representing the same concept $a = p$); $s^n$ is a negative term for $s^n$ (i.e., $s^n$ and $s^n$ represent non-matching concepts). Each triple produces a positive pair $(s^n, s^p)$ and a negative pair.
(s^a, s^n). To keep only the most informative triples, we use online hard mining for valid triplets in respect to the following constraint:

$$\| f_{enc}(s^a) - f_{enc}(s^p) \| < \| f_{enc}(s^a) - f_{enc}(s^n) \| + \lambda$$

where $f_{enc}$ is a BERT-based textual encoder, $\| \cdot \|$ is the normalized $L_2$-norm, and $\lambda$ is a pre-defined mining margin. Thus, the mining procedure discards all the triplets such that the distance from an anchor to its negative sample is greater than the distance to its positive sample by more than $\lambda$. Let $\mathcal{P}$ and $\mathcal{N}$ denote the sets of all positive and negative term pairs, respectively. The SAP procedure utilizes the Multi-Similarity (MS) loss (Wang et al., 2019) to learn from $\mathcal{P}$ and $\mathcal{N}$.

$$\mathcal{L}_{s_{ap}} = \frac{1}{|B|} \sum_{i=1}^{|B|} \left( \frac{1}{\alpha} \log \left( 1 + \sum_{n \in \mathcal{N}_i} e^{\alpha(S_{n} - \epsilon)} \right) + \frac{1}{\beta} \log \left( 1 + \sum_{p \in \mathcal{P}_i} e^{-\beta(S_{p} - \epsilon)} \right) \right)$$

where $\alpha$, $\beta$, and $\epsilon$ are the parameters of MS-loss. $\mathcal{P}_i$ and $\mathcal{N}_i$ are the sets of positive and negative samples for the anchor concept $i$. $S_{n}$ and $S_{p}$ are the cosine similarities of anchor $i$ to negative sample $n$ and positive sample $p$, respectively.

### 3.3 Graph neural networks

#### 3.3.1 Message passing layers

A common approach to capture the complex relationships between nodes in the graph is to iteratively update the representation of a node $v$ by passing and aggregating messages from its local node neighborhood $N(v)$ using a graph neural network. Gilmer et al. (2017) proposed a general Message Passing Neural Network (MPNN), which applies a composition of message function $f_m$ and $f_u$ to update the node representation $h_v^{(l)}$ at the $(l+1)$-th MPNN layer:

$$h_v^{(l+1)} = f_u(h_v^{(l)}, \sum_{(r,u) \in N(v)} f_m(h_v^{(l)}, h_u^{(l)}, e_r))$$

where $N(v) = \{(r,u) | (v, r, u) \in E\}$ is the set of pairs describing the local neighborhood of node $v$; $e_r$ are the edge features. Each pair $(r, u)$ indicates the presence of a directed edge of type $r$ from node $v$ to node $u$ in graph $G$. To avoid excessive computational complexity caused by significant variation in the number of neighbors across different nodes, we use a uniformly drawn fixed-size subset of neighbors instead of the entire node neighborhood as proposed by Hamilton et al. (2017). The primary distinguishing factor among various GNN models is the selection of $f_m$ and $f_u$ functions in the MPNN computational block. In GraphSAGE (Hamilton et al., 2017), a common and rather simple implementation of MPNN framework, an element-wise operator (e.g., max- or mean-pooling) is used as an $f_m$ to aggregate the vectors of neighbor nodes $N(v)$ into a single vector. The aggregated representation is further concatenated with the original representation and passed to a linear layer $W^{l+1}$ with a non-linear activation function $\sigma$. In this work, we use the GraphSAGE implementation with mean-pooling aggregation:

$$h_v^{(l+1)} = \sigma(W^l \cdot [h_v^{(l)} \parallel MEAN(N(v))])$$

where $MEAN$ is the mean-pooling operator, $[\cdot \parallel \cdot]$ is the concatenation of two vectors. Since the parameter matrix $W^l$ is the same for each $r \in R$ GraphSAGE operator prevents a thorough use of edge types and features. Regarding the UMLS graph, it means that GraphSAGE can only capture the textual node features and knowledge graph geometrical structure, though it doesn’t consider relation-specific information.

Schlichtkrull et al. (2018) proposed Relational Graph Convolutional Network (R-GCN) architecture that performs relation-aware message passing by introducing a relation-specific parameter matrix $W^l_r$ for each relation $r \in R$ at the neighborhood aggregation step of Message Passing block:

$$h_v^{(l+1)} = \sigma \left( \sum_{(r,u) \in N(v)} \frac{1}{|N(v)|} (W^l_r h_u^{(l)} + W^l_o h_v^{(l)}) \right)$$

$\sigma$ is a non-linear activation function and $W^l_o$ is a self-loop parameter matrix.

Despite being able to perform relation-aware message passing, R-GCN shares the limitation of GraphSAGE as it does not allow learning the relative importance of neighboring nodes. Graph attention network (GAT) (Veličković et al., 2018; Brody et al., 2022) addresses the limitation by introducing the self-attention over neighboring nodes and learning the aggregated neighborhood representation as the weighted sum of neighboring nodes representations. Given two node representations $h_{u}^{(l-1)}$ and $h_{v}^{(l-1)}$, the $l$-th GAT layer computes the relevance
of node \( u \) for the target node \( v \) as the normalized attention score \( \alpha_{uv}^{(l)} \):

\[
e^{(l)}_{uv} = a^T \cdot \text{LeakyReLU}(W^l : [h_{u}^{(l-1)} || h_{v}^{(l-1)}])
\]

\[
\alpha_{uv}^{(l)} = \frac{e^{(l)}_{uv}}{\sum_{(r,w) \in N(u)} e^{(l)}_{uw}},
\]

where \( a \) is a learnable weight vector. With the attention scores obtained, the aggregated neighborhood representation is computed as a weighted sum of neighboring nodes embeddings:

\[
h_v^{(l+1)} = \sigma \left( \sum_{(r,a) \in N(v)} \alpha_{uv} \cdot W^l h_u^{(l)} + W^l h_v^{(l)} \right)
\]

### 3.3.2 Deep Graph Infomax framework

To enrich an LM and a GNN with graph structure knowledge, we utilize the Deep Graph InfoMax framework (Veličković et al., 2019). DGI adopts an encoder model \( GNN \) to maximize the mutual information between global graph structure and its local subgraphs. This method is grounded on minimizing a noise-contrastive loss function

\[
\mathcal{L}_{dgi} = \frac{1}{N + M} \sum_{i=1}^{N} \text{GNN} \left[ \log D(\tilde{h}_i, \bar{s}) \right] + \sum_{j=1}^{M} \text{GNN} \left[ \log \left( 1 - D(\tilde{h}_j, \bar{s}) \right) \right],
\]

where \( \bar{s} \) is a graph summary obtained as a mean embedding of nodes in a graph; \( D \) is a learnable discriminator with sigmoid activation which scores summary-graph pairs. The loss aims at distinguishing between input node representations \( h \) and negative samples \( \tilde{h} \). The procedure starts with collecting initial and corrupted node features of graph \( G \), after that we obtain representations via \( GNN \), respectively, for each type of feature vectors. By leveraging local mutual information maximization and graph convolutional architectures, we make the features of local subgraphs to be consistent with the global properties of a larger graph.

### 3.4 BERGAMOT

We introduce a novel biomedical entity representation learning model BERGAMOT which leverages graph and textual encoders to infuse inter-concept interactions from a biomedical KG into LM. In BERGAMOT, we adopt and extend the pretraining procedure described in Sec. 3.2. Before moving on to the description of the pretraining procedure, let us first describe the structure of a training batch \( B \) which consists of (i) textual positive concept name pairs and (ii) local concept subgraphs.

#### Textual pairs

We begin by sampling a set \( T \), consisting of \( N \) random positive concept name pairs for concepts that have at least two distinct concept names. Let \((u, v)\) denote a pair of concept names that represent the same concept \( c \) (e.g., "Maux de tête" and "headache" as shown in Fig 1). We encode both textual names with a BERT-based LM resulting in two vector representations \( e^u_c = \text{LM}(u) \) and \( e^v_c = \text{LM}(v) \), respectively.

#### Concept subgraphs

A positive textual pair \((u, v)\) in \( T \) corresponds to a single UMLS concept \( c \). As \( c \) is also a node in the UMLS KG, we sample the graph \( G_c \) that is centered around \( c \) and includes a set of its neighboring nodes \( N(c) \). Next, we have two ways to initialize the central node \( c \) from \( G_c \) with either \( \text{LM}(u) \) or \( \text{LM}(v) \) resulting in two graphs \( G^n_c \) and \( G^v_c \) with identical structure but different initial node features. For instance, we can initialize a central node with a BERT embedding of either "headache" or "Maux de tête" (see Fig. 1). Non-central concept nodes are initialized with LM embeddings of their random concept names independently for both graphs. By applying a \( L \)-layer graph neural network we obtain two graph representations of the concept \( c \): \( g^n_c = \text{GNN}(G^n_c) \) and \( g^v_c = \text{GNN}(G^v_c) \).

BERGAMOT’s design is inspired by two major goals. First, we want to encode concept names and graph structure into a single shared embedding space while preserving an essential property for entity linking: various (both textual and graph) representations of the same concept must have similar embeddings. Second, since obtaining a relevant KG subgraph is not feasible at the inference stage, we want to infuse rich structural knowledge from the UMLS graph into a language model. To reach both goals, our model simultaneously learns 4 contrastive objectives on concept representations as shown in Fig. 1.

#### Term-term contrastive loss \( L_{sup} \)

To let a BERT-based model learn semantic and lexical similarity of various concept names, we apply MS-loss on textual pairs. This objective seeks to pull textual embeddings \((e^n_c, e^v_c)\) of concept \( c \)'s synonymous names closer in terms of cosine similarity, ignoring inter-concept relations. We use in-batch hard
### Table 1: Multilingual evaluation results in terms of acc@1 and acc@5 on the English, Spanish, Dutch, French, and German subsets of Mantra corpus. The best results are highlighted in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>English (acc@1)</th>
<th>English (acc@5)</th>
<th>Spanish (acc@1)</th>
<th>Spanish (acc@5)</th>
<th>Dutch (acc@1)</th>
<th>Dutch (acc@5)</th>
<th>French (acc@1)</th>
<th>French (acc@5)</th>
<th>German (acc@1)</th>
<th>German (acc@5)</th>
<th>avg (acc@1)</th>
<th>avg (acc@5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>Full test set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOTA</td>
<td>94.25</td>
<td>97.12</td>
<td>85.54</td>
<td>92.17</td>
<td>85.83</td>
<td>94.14</td>
<td>89.55</td>
<td>95.02</td>
<td>87.88</td>
<td>93.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Alekseev et al., 2022)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mSapBERT</td>
<td>90.04</td>
<td>96.90</td>
<td>83.67</td>
<td>92.17</td>
<td>84.25</td>
<td>92.43</td>
<td>93.69</td>
<td>94.06</td>
<td>92.08</td>
<td>89.15</td>
<td>88.55</td>
<td>92.98</td>
</tr>
<tr>
<td>mCODER</td>
<td>93.14</td>
<td>96.68</td>
<td>83.73</td>
<td>90.96</td>
<td>85.04</td>
<td>90.55</td>
<td>89.19</td>
<td>92.79</td>
<td>85.87</td>
<td>87.57</td>
<td>94.53</td>
<td>85.44</td>
</tr>
<tr>
<td>GraphSAGE-BERGAMOT</td>
<td>93.81</td>
<td>96.90</td>
<td>81.93</td>
<td>90.96</td>
<td>84.25</td>
<td>92.43</td>
<td>93.24</td>
<td>94.06</td>
<td>88.06</td>
<td>94.03</td>
<td>86.55</td>
<td>92.98</td>
</tr>
<tr>
<td>RGCN-BERGAMOT</td>
<td>93.36</td>
<td>96.46</td>
<td>83.73</td>
<td>92.17</td>
<td>84.25</td>
<td>92.43</td>
<td>93.24</td>
<td>94.06</td>
<td>88.06</td>
<td>94.03</td>
<td>86.55</td>
<td>92.98</td>
</tr>
<tr>
<td>GAT-BERGAMOT</td>
<td>94.69</td>
<td>97.57</td>
<td>85.54</td>
<td>94.58</td>
<td>86.61</td>
<td>94.14</td>
<td>91.54</td>
<td>97.01</td>
<td>88.61</td>
<td>95.52</td>
<td>86.55</td>
<td>93.14</td>
</tr>
</tbody>
</table>

**Filtered test set**

<table>
<thead>
<tr>
<th>Model</th>
<th>English (acc@1)</th>
<th>English (acc@5)</th>
<th>Spanish (acc@1)</th>
<th>Spanish (acc@5)</th>
<th>Dutch (acc@1)</th>
<th>Dutch (acc@5)</th>
<th>French (acc@1)</th>
<th>French (acc@5)</th>
<th>German (acc@1)</th>
<th>German (acc@5)</th>
<th>avg (acc@1)</th>
<th>avg (acc@5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>Supervised</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOTA</td>
<td>80.95</td>
<td>91.27</td>
<td>75.32</td>
<td>87.01</td>
<td>80.00</td>
<td>66.67</td>
<td>86.87</td>
<td>80.37</td>
<td>90.65</td>
<td>76.35</td>
<td>87.16</td>
<td></td>
</tr>
<tr>
<td>(Alekseev et al., 2022)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mSapBERT</td>
<td>80.16</td>
<td>90.48</td>
<td>71.43</td>
<td>87.01</td>
<td>75.38</td>
<td>80.0</td>
<td>62.63</td>
<td>85.86</td>
<td>77.57</td>
<td>91.59</td>
<td>73.43</td>
<td>86.99</td>
</tr>
<tr>
<td>mCODER</td>
<td>76.98</td>
<td>89.68</td>
<td>71.43</td>
<td>81.82</td>
<td>76.92</td>
<td>86.15</td>
<td>77.78</td>
<td>83.84</td>
<td>74.77</td>
<td>89.72</td>
<td>75.58</td>
<td>86.24</td>
</tr>
<tr>
<td>GraphSAGE-BERGAMOT</td>
<td>79.37</td>
<td>90.48</td>
<td>67.53</td>
<td>84.42</td>
<td>75.38</td>
<td>84.62</td>
<td>67.68</td>
<td>84.85</td>
<td>77.57</td>
<td>88.79</td>
<td>73.51</td>
<td>86.63</td>
</tr>
<tr>
<td>RGCN-BERGAMOT</td>
<td>77.78</td>
<td>88.89</td>
<td>71.43</td>
<td>85.71</td>
<td>75.38</td>
<td>84.62</td>
<td>69.70</td>
<td>83.84</td>
<td>76.64</td>
<td>92.52</td>
<td>74.19</td>
<td>87.12</td>
</tr>
<tr>
<td>GAT-BERGAMOT</td>
<td>82.54</td>
<td>92.86</td>
<td>75.32</td>
<td>90.91</td>
<td>80.00</td>
<td>84.62</td>
<td>67.68</td>
<td>86.87</td>
<td>84.11</td>
<td>94.39</td>
<td>76.78</td>
<td>89.93</td>
</tr>
</tbody>
</table>

### Table 2: Evaluation results of GAT-BERGAMOT models pre-trained on monolingual and multilingual graphs in terms of Acc@1 and Acc@5 on filtered test sets of the Mantra corpus. The best results are highlighted in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>Base model</th>
<th>Graph</th>
<th>Spanish (acc@1)</th>
<th>Spanish (acc@5)</th>
<th>Dutch (acc@1)</th>
<th>Dutch (acc@5)</th>
<th>French (acc@1)</th>
<th>French (acc@5)</th>
<th>German (acc@1)</th>
<th>German (acc@5)</th>
<th>avg (acc@1)</th>
<th>avg (acc@5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
<td>@1</td>
<td>@5</td>
</tr>
<tr>
<td>mSapBERT</td>
<td>mSapBERT</td>
<td>—</td>
<td>71.43</td>
<td>87.01</td>
<td>75.38</td>
<td>80.0</td>
<td>62.63</td>
<td>85.86</td>
<td>77.57</td>
<td>91.59</td>
<td>71.75</td>
<td>86.12</td>
</tr>
<tr>
<td>SapBERT text only</td>
<td>mSapBERT</td>
<td>—</td>
<td>67.53</td>
<td>88.31</td>
<td>76.92</td>
<td>83.08</td>
<td>71.72</td>
<td>88.89</td>
<td>84.11</td>
<td>92.52</td>
<td>75.07</td>
<td>88.2</td>
</tr>
<tr>
<td>BERGAMOT</td>
<td>mSapBERT</td>
<td>Monolingual</td>
<td>71.43</td>
<td>83.12</td>
<td>80.0</td>
<td>83.08</td>
<td>69.70</td>
<td>85.86</td>
<td>80.37</td>
<td>92.52</td>
<td>75.38</td>
<td>86.15</td>
</tr>
<tr>
<td></td>
<td>XLMR</td>
<td>Multilingual</td>
<td>75.32</td>
<td>90.91</td>
<td>80.0</td>
<td>84.62</td>
<td>67.68</td>
<td>86.87</td>
<td>84.11</td>
<td>94.39</td>
<td>76.78</td>
<td>89.2</td>
</tr>
</tbody>
</table>

### Node-node contrastive loss \( L_{\text{node}} \)

Similarly to term-term loss, we apply MS-loss to pull graph embeddings \((g_c^u, g_c^v)\) representing the same concept \(c\) closer pushing away the representations of non-matching ones. Unlike textual encoder, GNN is aware of relations between concepts as it sees neighboring nodes \(N(c)\) rather than a single concept \(c\) only.

### Term-node contrastive loss \( L_{\text{int}} \)

In batch \(B\), a central concept node \(c\) has four representations: \(e_c^u\), \(e_c^v\), \(g_u^c\), and \(g_v^c\). While \(L_{\text{sap}}\) and \(L_{\text{node}}\) optimize unimodal textual and graph models separately, our ultimate goal is to enhance a language model with graph structure knowledge accumulated in graph embeddings. To enable mutual information sharing between a text and a graph encoder we introduce a third contrastive loss which learns from two-modal positive pairs \((e_u^c, g_u^c)\) and \((e_v^c, g_v^c)\). Intuitively, we expect our LM to memorize in-domain knowledge from KG. Similarly to term-term and node-node objectives, we adopt MS-loss and two-modal in-batch hard negative samples. Since \(L_{\text{int}}\) makes LM and graph encoders exchange their knowledge, we encourage an LM to be more aware of inter-concept relations.

### DGI loss

As an additional pretraining objective for a graph encoder, we employ the DGI framework and calculate the DGI loss \(L_{\text{dgi}}\). We use a union of positive samples to push textual representations of non-matching concepts far from each other.
of all local concept subgraphs from the batch $B$ to form a global batch $G_B$. Readout function $R(G_B)$ calculates a mean embedding over $N$ central nodes of concept subgraphs. To obtain a corrupted graph $\tilde{G}_B$, we randomly shuffle central nodes across all positive paired samples. The choice of the corruption function encourages our model to distinguish if nodes $N(c)$ are actual neighbors of a central node $c$. The intuition behind adding the DGI loss is to learn more informative node features. Since DGI captures graph structural information by learning on true and corrupted graphs, we expected it to contribute to node representation. The intuition behind adding the DGI loss is to learn more informative node features. Since DGI captures graph structural information by learning on true and corrupted graphs, we expected it to contribute to the model's ability to distinguish if nodes $N(c)$ are actual neighbors of a central node $c$.

For $L_{	ext{sap}}$, $L_{	ext{node}}$, and $L_{	ext{int}}$ we employ MS-loss (see Eq. 1) for maximum comparability with multilingual versions of SapBERT (Liu et al., 2021b) and CODER (Yuan et al., 2022) as both models utilize this contrastive objective in $L_{	ext{sap}}$. Also, MS-loss was experimentally shown to be the most effective contrastive objective to learn from UMLS synonyms (Liu et al., 2021a) among 7 variants. The final training objective used to pre-train the BERGAMOT model is defined as the weighted sum of four loss functions:

$$L = L_{\text{sap}} + L_{\text{node}} + L_{\text{int}} + \lambda_{\text{dgi}} L_{\text{dgi}},$$

where $\lambda_{\text{dgi}}$ is the pre-selected weight of DGI loss. After pretraining on UMLS we discard the graph encoder as in general case there is no graph available during inference. Thus, the result of the pre-training procedure is a BERT-based model enriched with knowledge from the UMLS graph.

4 Experimental evaluation

To train BERGAMOT, we use the UMLS 2020AB release which contains approximately 4.4 million concepts and 15.9 million unique concept names from 215 source vocabularies. The full statistics on pre-training multilingual and monolingual data as well as the number of concept names for languages are shown in Appendix A. We remove all duplicated edges (i.e., edges with matching source and target concepts, and relation type). To ensure each batch includes a sufficient amount of positive samples, we pre-generate synonymous concept name pairs following Liu et al. (2021a). Since random sampling could result in the underrepresentation of languages other than English, we explicitly add (i) monolingual non-English and (ii) cross-lingual positive concept name pairs to each batch. However, the quadratic growth in the number of possible language pairs with respect to the number of languages limits our ability to enrich the training dataset with multilingual data. We discard all cross-lingual pairs which consist of two non-English terms thus forcing all languages to benefit from extensive English knowledge.

To explore how the BERGAMOT’s performance is affected by a monolingual low-resource setting, we additionally subsample a monolingual subgraph and generate a monolingual set of positive pairs for Spanish, Dutch, French, and German. We create each dataset by removing all terms that came from a non-target language and all nodes that have no concept name in a target language. For statistics on positive textual pairs count, number of edges and nodes, please see Tab. 8 in Appx B.

For evaluation on the MCN task, we use two cross-lingual benchmarks:

- a medical-crossing benchmark (Alekseev et al., 2022) based on Mantra corpus (Kors et al., 2015) of text units such as scientific abstract titles, drug labels, patent claims mapped to the UMLS concepts.
- XL-BEL (Liu et al., 2021b), with Wikipedia entities linked to the UMLS.

Mantra corpus covers mentions in English, French, German, Spanish, and Dutch while XL-BEL covers 10 languages. Both benchmarks allow zero-shot evaluation only, i.e., there are no training sets. Additionally, we employ the French Quaero corpus (Névéol et al., 2014) and two datasets in Spanish: (i) CodiEsp-Diagnostico (Miranda-Escalada et al., 2020b) and (ii) CANTEMIST (Miranda-Escalada et al., 2020a). Alekseev et al. (2022) introduced a novel filtering procedure which drops test set mentions that are identical to a term from the dictionary. For dataset descriptions and statistics, refer to Appendix B.

To explore the ability of our model to solve diverse non-MCN NLP tasks, we additionally evaluate our model on biomedical Question Answering (QA) and Textual entailment (TE) tasks. We conducted QA experiments on two datasets: (i) PubMedQA (Jin et al., 2019) and BioASQ (Nentidis et al., 2019). The goal of TE task is to determine a logical relationship between two pieces of text: a premise and a hypothesis. For TE experiments, we utilize two corpora: (i) MedNLI (Shivade et al., 2019) and (ii) SciTail (Khot et al., 2018). For
Table 3: Evaluation results in terms of acc@1 and acc@5 on filtered test sets of the French QUAERO corpus (EMEA and MEDLINE subsets) and the Spanish CodiEsp Diagnostico and CANTEMIST corpora. The best results are highlighted in bold. † denotes statistical significance of GAT-BERGAMOT over both mSapBERT and CODER (Wilcoxon, $\rho < 0.01$).

Table 4: Evaluation of BERGAMOT model with GAT graph encoder on multilingual XL-BEL benchmark. The benchmark includes entities in English (en), Spanish (es), German (de), Finnish (fi), Russian (ru), Turkish (tr), Korean (ko), Chinese (zh), Japanese (jp), Thai (th) languages. The best results are highlighted in bold.

Table 5: Evaluation of GAT-BERGAMOT on: (i) Question Answering PubMedQA (PMQA) and BioASQ datasets; (ii) Textual Entailment MedNLI and SciTail (ST) datasets in terms of accuracy.

Table 6: Ablation results of GAT-BERGAMOT. We report the mean acc@1 and acc@5 over all languages present in the Mantra and XL-BEL corpora. $L_{\text{sap}}$ only set-up refers to mSapBERT which trains with a single text-based training objective.

4.1 Experimental Setup

For evaluation, we employ a ranking approach over embeddings of mentions and potential concepts. After applying an average pooling layer over a BERT-based encoder, the inference task is then reduced to finding the closest concept name representation to the entity mention representation in a joint embedding space. We use the Euclidean distance as the metric. Nearest concept names are chosen as top-$k$ concepts for entities. We evaluate the models in the information retrieval scenario, where the goal is to find top-$k$ concepts for every entity mention in a dictionary of concept names and their identifiers. Following previous works on entity linking (Suominen et al., 2013; Pradhan et al., 2014; Wright et al., 2019; Phan et al., 2019; Liu et al., 2021a), we use the top-$k$ accuracy as the evaluation metric: $\text{Acc}@k = 1$ if the correct UMLS concept unique identifier is retrieved at the rank $\leq k$, otherwise $\text{Acc}@k = 0$. We note that BERGAMOT’s graph encoder is discarded during inference and only a BERT-based encoder is used for ranking.

4.2 Results

**Medical Concept Normalization** Tab. 1 shows the acc@1 and acc@5 metrics for datasets in five languages. BERGAMOT outperformed all models on four sets except the French dataset on both full and filtered test sets. The best results are achieved by GAT-BERGAMOT which consistently ourper-
forms mSapBERT on all languages proving the effectiveness of three additional training objectives that rely on graph embeddings. Poor performance of GraphSAGE- and RGCN- versions of our model indicate the effectiveness of relative neighbor importance learnt in GAT via the attention mechanism. Interestingly, GAT-BERGAMOT achieves state-of-the-art results on both resource-rich (English and Spanish) and low-resource (German and Dutch) languages. On English, it outperforms a supervised model that is fine-tuned on English data. We must note that due to the small dataset size, GAT-BERGAMOT’s improvement over mSapBERT and mCODER is statistically significant (Wilcoxon, $\rho < 0.05$) for the German part only. We provide examples of models’ predictions in Appx. G.

In the next series of experiments, we explored how BERGAMOT benefits from both monolingual and multilingual pre-training set-ups. For each of four non-English parts of the Mantra corpus, we trained a monolingual SapBERT and a monolingual GAT-BERGAMOT model. Tab. 2 presents the performance of models pre-trained on multilingual and non-English monolingual graphs and concept names. Based on the results, we can notice that BERGAMOT benefits from the multilingual bi-modal data the most. It appears that training a single multilingual graph-augmented BERGAMOT eliminates a need for further pretraining of monolingual LMs (either SapBERT or BERGAMOT) on UMLS. On average, monolingual BERGAMOT models perform on par with monolingual SapBERT models.

Tab. 3 presents the evaluation results on filtered test sets of the French QUAERO corpus and the Spanish CodiEsp-D and CANTEMIST corpora. GAT-BERGAMOT consistently outperforms two other BERGAMOT models as well as mSapBERT and mCODER baselines pushing the existing state-of-the-art on these corpora. Notably, the GAT-BERGAMOT’s improvement over both baselines is statistically significant (Wilcoxon, $\rho < 0.01$). While mCODER outperforms GAT-BERGAMOT on the French part of Mantra, the latter shows a statistically significant improvement on both subsets of QUAERO corpus.

We further investigated the BERGAMOT’s performance on the XL-BEL benchmark. Tab. 4 presents the evaluation results. GAT-BERGAMOT outperforms mSapBERT on 8 of 10 languages with an average improvement of 0.8% acc@1. The largest acc@1 growth compared to mSapBERT is observed for low-resource languages (+4.1% for fi, +1.9% for ja, +1.3% for ru). Since mCODER does not support Korean and Thai, it is not directly comparable to GAT-BERGAMOT on full XL-BEL. However, our model outperforms mCODER on 8 remaining languages.

The evaluation results of GAT-BERGAMOT on question answering and textual entailment tasks are presented in Table 5. Despite an introduction of additional entity-oriented graph-based losses, GAT-BERGAMOT did not lose the capability to solve tasks that require text understanding. It performs on par or better than text-only SapBERT. An improvement of 2.1% over SapBERT on PubMedQA indicates a potential of using graph-induced pretraining objectives for tasks involving domain-specific knowledge, such as question answering.

Ablation study To explore the effectiveness of each training objectives, we conducted an ablation analysis by training a GAT modification with each of four individual training objectives removed. Tab. 6 shows the change in performance on XL-BEL and the filtered version of Mantra. Despite losing 1.5% and 1.68% acc@1 on Mantra and XL-BEL, respectively, a model with no $L_{dgi}$ still shows a decent performance outperforming SapBERT on Mantra. With DGI loss removed, BERGAMOT still outperforms both SapBERT (+2.82%) and CODER (+0.67%). Removal of explicit modality interaction introduced through $L_{int}$ results in a slight drop of 0.52% and 0.7% acc@1 on Mantra and XL-BEL, respectively. Interestingly, removal of $L_{node}$ leads to an average 3.38% acc@1 drop on Mantra and an average improvement of 0.5% on XL-BEL.

5 Conclusion and Future Work

We presented BERGAMOT, a graph-augmented architecture with backbone LM designed to learn inter-concept and intra-concept interactions from the multilingual knowledge graph. BERGAMOT outperforms existing language models pre-trained on knowledge triples from UMLS on two concept normalization benchmarks with a diverse set of languages. Since BERGAMOT is currently an encoder model and it is not able to generate texts, an important future research would be to extend BERGAMOT to include language generation capabilities and advance KG-enhanced language generation.
Acknowledgments

The work was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Ivanivkov Institute for System Programming of dated November 2, 2021 No. 70-2021-00142. We specially thank Zulfat Miftahutdinov for useful discussions and feedback pointing out issues in a previous iteration of the paper. We thank anonymous reviewers for valuable feedback.

6 Limitations

Large domain-specific graphs. The graph neural networks in BERGAMOT employ a large biomedical knowledge graph, the Unified Medical Language System (UMLS), which contains over 4 million concepts and 15 million concept names. It is important to note that the use of knowledge graphs for different domains with a smaller number of nodes and edges may affect the performance. The size and complexity of the knowledge graph can have a significant impact on the ability of the model to learn and make accurate predictions. Additionally, it is worth noting that while the study focused on the biomedical domain, BERGAMOT could potentially be trained on general-domain knowledge graphs, such as DBPedia or Wikidata.

Only entity-related tasks for evaluation. In this paper, experiments were done on cross-lingual entity linking to see how well BERGAMOT captures knowledge of a multilingual knowledge graph in the biomedical domain. However, to fully assess the ability of BERGAMOT to facilitate automatic knowledge base construction in different languages, it may be beneficial to include additional evaluation tasks such as link prediction or candidate-free taxonomy enrichment. Additionally, probing the knowledge contained in BERGAMOT across multiple languages could be evaluated using the multilingual Language Model Analysis (LAMA) benchmark.

Mantra and XL-BEL. We acknowledge that our choice of Mantra and XL-BEL datasets for MCN is non-exhaustive: we focus on standard benchmarks from previous work, yet both datasets have certain limitations. For example, the Mantra dataset is manually annotated but relatively small, which can make it difficult to measure statistical significance. Similarly, the XL-BEL dataset is based on Wikipedia texts without manual annotation by medical experts, which may limit its relevance to real-world medical applications. Furthermore, both datasets only include entity mentions and terminology in a target language (e.g., French), while BERGAMOT can encode concept names in as many languages as possible for inference to better exploit language connections.

Ethics Statement

One limitation of using external knowledge sources, such as the Unified Medical Language System (UMLS), is that these sources may not be complete for all languages, which can affect the performance of language models and their ability to infer medical concepts from text. Additionally, significant changes to UMLS may require re-training of the language model. BERGAMOT, like any language model, may be subject to representation biases and potentially misleading results, which is a critical concern in the healthcare domain.

All pre-trained models, UMLS, Mantra, XL-BEL, BioASQ, PubMedQA, MedNLI, and SciTail datasets used in this work are publicly available for research purposes.

We honor and support the ACL Code of Ethics.

References


Alan R. Aronson. 2001. Effective mapping of biomedical text to the UMLS metathesaurus: the metamap


Kirk Roberts, Dina Demner-Fushman, and Joseph M Tonning. 2017. Overview of the tac 2017 adverse reaction extraction from drug labels track. In TAC.


A  Statistics on UMLS graph

Tab. 7 presents statistics on the number of concept names for each language.

The statistics on pre-training graph and number of positive concept name pairs is shown in Tab. 8. The most resource-rich languages in the biomedical domain are English and Spanish, while German and Dutch can be considered as low-resource languages. In particular, the German UMLS is only 1.48% of the UMLS in vocabulary and 3.72% in source counts (NIH).

B  Concept Normalization data

Mantra GSC (Kors et al., 2015) is a collection of biomedical text units such as drug labels and patent claims manually cross-labeled by several annotators in five different languages: English, French, German, Spanish, and Dutch. The Mantra terminology is a subset of UMLS with concepts from MeSH, SNOMED-CT, and MedDRA extracted from the UMLS 2020 AA release. Since the Mantra corpus is too small for fine-tuning, all entity mentions are used as test data.

The CodiEsp dataset was presented at Clinical Case Coding in Spanish Shared Task at the CLEF 2020 evaluation forum (Miranda-Escalada et al., 2020b). It contains clinical records with entities mapped to the ICD-10 vocabulary (CodeBooks, 2016); we use the CodiEsp Diagnosis (CodiEsp-D) subset and the dictionary provided in CodiEsp.

CANTEMIST (CANcer TExt MIning Shared Task on IberLEF 2020 (Miranda-Escalada et al., 2020a) is a manually annotated text corpus of tumor morphology mentions in Spanish mapped to the latest Spanish version of the oncological ontology, which is a part of ICD-O (World Health Organization, 2013); we use the dictionary from (López-Úbeda et al., 2020).

The QUAERO French Medical Corpus (Névédol et al., 2014) is collection of French entities from two categories: (i) information on marketed drugs from the European Medicines Agency (EMEA, 12,761 entities) and (ii) titles of research articles indexed in the MEDLINE database (8,781 entities). Test set filtering reduces the number of entities to 5,533 and 3,534 for EMEA and MEDLINE parts, respectively. We use the French concept names from the UMLS as a dictionary.

XL-BEL (Liu et al., 2021b) is an automatically
annotated dataset with concept mentions from Wikipedia articles. Each mention is mapped to a Wikipedia article using a hyperlink and assigned a CUI based on the article’s metadata. Each language is represented with 1,000 mentions resulting in 10,000 mentions in total.

As shown in Table 9, there are 3 evaluation types (Alekseev et al., 2022):

- **Full**: compute metrics on the test set as provided in the dataset itself;
- **Filtered**: remove from the set all entities that are already present in the dictionary (exact match);
- **Filtered$_{0.2}$**: remove from the set all entities where the character-based Levenshtein distance to the nearest neighbor in the dictionary is under 0.2.

The filtered$_{0.2}$ set contains two times fewer entities compared to the filtered set. Hence, we chose the full and filtered sets for experiments. All sets, dictionaries, and evaluation scripts are available at https://github.com/AIRI-Institute/medical_crossing.

Table 9: Statistics of multilingual corpora user for the evaluation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># in full set</th>
<th>Avg. len in chars</th>
<th>% with numerals</th>
<th>Filtered set</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entity mentions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mantra (de)</td>
<td>201</td>
<td>17.62</td>
<td>0.50</td>
<td>107</td>
</tr>
<tr>
<td>Mantra (en)</td>
<td>452</td>
<td>16.42</td>
<td>1.11</td>
<td>126</td>
</tr>
<tr>
<td>Mantra (es)</td>
<td>166</td>
<td>19.67</td>
<td>2.41</td>
<td>99</td>
</tr>
<tr>
<td>Mantra (fr)</td>
<td>222</td>
<td>16.06</td>
<td>0.00</td>
<td>65</td>
</tr>
<tr>
<td>Mantra (nl)</td>
<td>127</td>
<td>16.06</td>
<td>0.00</td>
<td>65</td>
</tr>
<tr>
<td>CANTEMIST</td>
<td>10031</td>
<td>18.73</td>
<td>6.92</td>
<td>3268</td>
</tr>
<tr>
<td>CodiEsp-D</td>
<td>10874</td>
<td>15.84</td>
<td>1.05</td>
<td>3449</td>
</tr>
<tr>
<td><strong>Concepts</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mantra (de)</td>
<td>169</td>
<td>-</td>
<td>-</td>
<td>97</td>
</tr>
<tr>
<td>Mantra (en)</td>
<td>373</td>
<td>-</td>
<td>-</td>
<td>119</td>
</tr>
<tr>
<td>Mantra (es)</td>
<td>147</td>
<td>-</td>
<td>-</td>
<td>69</td>
</tr>
<tr>
<td>Mantra (fr)</td>
<td>185</td>
<td>-</td>
<td>-</td>
<td>83</td>
</tr>
<tr>
<td>Mantra (nl)</td>
<td>117</td>
<td>-</td>
<td>-</td>
<td>62</td>
</tr>
<tr>
<td>CANTEMIST</td>
<td>657</td>
<td>-</td>
<td>-</td>
<td>364</td>
</tr>
<tr>
<td>CodiEsp-D</td>
<td>2206</td>
<td>-</td>
<td>-</td>
<td>1142</td>
</tr>
</tbody>
</table>

C Hyperparameter settings

A list of hyperparameter values used to train BERGAMOT models is presented in Table 10. We adopted the parameters of the MS-loss from SapBERT for fair comparison. For the DGI weight $\lambda_{dgi}$, we found the best value from the list (0.01, 0.1, 1.0).

Table 10: BERGAMOT’s hyperparameter values

<table>
<thead>
<tr>
<th>Hyperparameter name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph encoder hidden size</td>
<td>768</td>
</tr>
<tr>
<td>Number of neighbors</td>
<td>3</td>
</tr>
<tr>
<td>Number of graph encoder layers</td>
<td>3</td>
</tr>
<tr>
<td>GAT’s number of attention heads</td>
<td>2</td>
</tr>
<tr>
<td>Weight $\lambda_{dgi}$</td>
<td>0.1</td>
</tr>
<tr>
<td>Hard pairs miner’s margin</td>
<td>0.2</td>
</tr>
<tr>
<td>$\alpha$ in MS loss</td>
<td>2</td>
</tr>
<tr>
<td>$\beta$ in MS loss</td>
<td>50</td>
</tr>
<tr>
<td>$\epsilon$ in MS loss</td>
<td>0.5</td>
</tr>
<tr>
<td>Learning rate</td>
<td>$2 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>Multilingual models’ batch size</td>
<td>256</td>
</tr>
<tr>
<td>Monolingual models’ batch size</td>
<td>128</td>
</tr>
<tr>
<td>Multilingual models’ # of epochs</td>
<td>1</td>
</tr>
<tr>
<td>Monolingual models’ # of epochs</td>
<td>2</td>
</tr>
</tbody>
</table>

Tab. 10 lists BERGAMOT’s hyperparameter values used throughout our experiments.

D Question Answering and Textual Entailment data

**BioASQ** BioASQ (Biomedical Question Answering) is a widely recognized dataset in the biomedical domain, specifically designed for evaluating question answering systems. Following (Gu et al., 2022), we restrict the dataset to yes/no questions. We use the official train/dev/test split with 670/75/140 questions respectively.

**PubMedQA** Similar to BioASQ, the PubMedQA dataset as well presents questions with limited number of answers. In contrast to the previous dataset, the answers to the questions in PubMedQA are selected from yes, no, or maybe. We use the original train/dev/test split with 450, 50, and 500 questions, respectively.

**MedNLI** MedNLI (Medical Natural Language Inference) is a specialized dataset designed to facilitate research in natural language inference within the medical and healthcare domain. It consists of pairs of sentences, where each pair comprises a premise and a hypothesis. The premise represents a clinical or biomedical context, while the hypothesis is a medical statement or claim that may or may not logically follow from the premise. Each sentence pair is annotated with one of three labels:
"entailment," indicating that the hypothesis can be logically inferred from the premise; "contradiction," suggesting that the hypothesis contradicts the information in the premise; and "neutral," signifying that there is no logical relationship between the two sentences. The dataset comprises a total of 12,627 sentence pairs in the training set and 1,422 sentence pairs in the testing set.

SciTail The SciTail dataset is similar to the MedNLI dataset was designed for the task of natural language inference. Except that it covers a broader scientific domain. The train part of the corpora contains 24900 sentence pairs and the test part of the corpora contains 2126.

E Models for Comparison

- XLM-R (Roberts et al., 2017) – checkpoint xlm-roberta-base (the Hugging Face Hub)
- SapBERT (Liu et al., 2021a,b) – SapBERT-UMLS-2020AB-all-lang-from-XLMR (the Hugging Face Hub)
- CODER (Yuan et al., 2022) – coder_all (the Hugging Face Hub)
- Supervised SOTA (Alekseev et al., 2022) – the results of SapBERT+mcn-fz4 are directly obtained from the authors’ paper. Three versions were fine-tuned using SapBERT-UMLS-2020AB-all-lang-from-XLMR: SapBERT+mcn-fz4, SapBERT+mcn-fz10, and SapBERT+mcn. The first one shows slightly better results in their paper.

F Hardware and software specification

Table 11: Hardware specification of the machine used to conduct our experiments

<table>
<thead>
<tr>
<th>Device</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>8x Intel Xeon Gold 6152 2.1-3.7 Ghz (2 cores each)</td>
</tr>
<tr>
<td>GPU</td>
<td>4x NVIDIA Tesla V100 32GB</td>
</tr>
<tr>
<td>RAM</td>
<td>768 Gb</td>
</tr>
<tr>
<td>Disk space</td>
<td>5 Tb</td>
</tr>
</tbody>
</table>

To implement, train, and evaluate our models, we used the version 1.11.0 of PyTorch (Paszke et al., 2019) with CUDA 11.3 (Nickolls et al., 2008) support. To implement graph neural networks, we used PyTorch Geometric (Fey and Lenssen, 2019) version 2.0.4. The training of each multilingual BERGAMOT model took up to 20 hours on the machine with the hardware specification described in Table 11.

G Examples of predictions

We provide a few examples of SapBERT’s and BERGAMOT’s predictions on the English Mantra in Table 12. The results show that SapBERT outperforms BERGAMOT when a true concept in a vocabulary has an extensive textual name. When a golden concept has a short name only, text-only SapBERT fails to produce good entity representations, which is not the case for BERGAMOT. Thus, the provided examples let us suggest that BERGAMOT does not only capture semantic and lexical similarity between entity and concept strings but also has additional intuition of what is a concept from the UMLS knowledge base.
<table>
<thead>
<tr>
<th>Mention</th>
<th>SapBERT prediction</th>
<th>BERGAMOT prediction</th>
<th>True concept</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>mixed collagen disease</td>
<td>collagen dis</td>
<td>connective tissue dis mixed</td>
<td>connective tissue dis mixed</td>
<td>BERGAMOT</td>
</tr>
<tr>
<td>uveitic</td>
<td>ectatic qualifier value</td>
<td>uveitis disorder</td>
<td>uveitis disorder</td>
<td>BERGAMOT</td>
</tr>
<tr>
<td>double uterus</td>
<td>uterus didelphus disorder</td>
<td>doubling of uterus nos disorder</td>
<td>doubling of uterus nos disorder</td>
<td>BERGAMOT</td>
</tr>
<tr>
<td>hygiene</td>
<td>ability to perform personal hygiene activity observable entity</td>
<td>personal hygiene finding procedure</td>
<td>ability to perform personal hygiene activity observable entity</td>
<td>SapBERT</td>
</tr>
<tr>
<td>functional bowel disorders</td>
<td>functional disorder of intestine disorder</td>
<td>x psychogenic ibs</td>
<td>functional disorder of intestine disorder</td>
<td>SapBERT</td>
</tr>
</tbody>
</table>

Table 12: Several SapBERT's and BERGAMOT's predictions on the English Mantra.