SN Computer Science (2022) 3:387
https://doi.org/10.1007/s42979-022-01295-7

ORIGINAL RESEARCH

Enhancing Cross-lingual Biomedical Concept Normalization Using Deep Neural Network Pretrained Language Models

Ying-Chi Lin[1] · Phillip Hoffmann[1] · Erhard Rahm[1]

Received: 13 October 2021 / Accepted: 30 June 2022

Abstract

In this study, we propose a new approach for cross-lingual biomedical concept normalization, the process of mapping text in non-English documents to English concepts of a knowledge base. The resulting mappings, named as semantic annotations, enhance data integration and interoperability of documents in different languages. The US FDA (Food and Drug Administration), therefore, requires all submitted medical forms to be semantically annotated. These standardized medical forms are used in health care practice and biomedical research and are translated/adapted into various languages. Mapping them to the same concepts (normally in English) facilitates the comparison of multiple medical studies even cross-lingually. However, the translation and adaptation of these forms can cause them to deviate from its original text syntactically and in wording. This leads the conventional string matching methods to produce low-quality annotation results. Therefore, our new approach incorporates semantics into the cross-lingual concept normalization process. This is done using sentence embeddings generated by BERT-based pretrained language models. We evaluate the new approach by annotating entire questions of German medical forms with concepts in English, as required by the FDA. The new approach achieves an improvement of 136% in recall, 52% in precision and 66% in F-measure compared to the conventional string matching methods.

Keywords Biomedical concept normalization · BERT · Sentence embedding · Cross-lingual · UMLS

Introduction

Concept normalization, also named as semantic annotation or entity linking, aims to map a sequence of text to a concept of a given knowledge base, such as an ontology, taxonomy or thesaurus. Those mappings or *annotations* have been applied to enhance search engines, data integration or drug discovery. For example, the MEDLINE database contains journal citations and abstracts for biomedical literature. The data in MEDLINE are annotated using the MeSH (Medical Subject Headings) vocabulary. PubMED,[1] the search engine accessing the MEDLINE database, uses these annotations to improve retrieval speed and quality.

Semantic annotations enhance interoperability of the documents and facilitates data integration. The CDISC Standards, jointly developed by the US Food and Drug Administration (FDA) and the Clinical Data Interchange Standards Consortium (CDISC), define the baselines of the interchange format of medical research data. Since 2016, regulatory submissions to the FDA such as new drug applications have to comply with those standards, that incorporate semantic annotation of any submitted medical form. These study data standards ensure the FDA to process the submissions more efficiently. Furthermore, they also facilitate the FDA to solve research questions that need to integrate data from multiple studies. The vocabulary used for the annotations are defined in the Study Data Tabulation Model Controlled Terminology (SDTM-CT), which is maintained and distributed as part of the NCI Thesaurus. This terminology covers a large set of medical forms, clinical studies and questionnaires,

This article is part of the topical collection "Biomedical Engineering Systems and Technologies" guest edited by Hugo Gamboa and Ana Fred.

✉ Ying-Chi Lin
lin@informatik.uni-leipzig.de

Phillip Hoffmann
ph30gabo@studserv.uni-leipzig.de

Erhard Rahm
rahm@informatik.uni-leipzig.de

[1] ScaDS.AI/Department of Computer Science, Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany

[1] PubMed https://www.ncbi.nlm.nih.gov/pubmed.

Published online: 21 July 2022

Question 1	
OE	Poor appetite or overeating
DE	Verminderter Appetit oder übermäßiges Bedürfnis zu essen
GO	Decreased appetite or excessive need to eat

Question 2	
OE	Tension
DE	Anspannung
GO	Tension

Associated UMLS concepts		
CUI	Concept Name	Form
C2706943	Poor appetite or overeating in last 2W.presence:^Patient:Ord:Observed	PHQ-9
C2706945	Poor appetite or overeating in last 2W.frequency:Patient:Ord:Observed	PHQ-9
C2707461	Poor appetite or overeating in last 2W.presence:^Patient:Ord:Reported	PHQ-9
C2707462	Poor appetite or overeating in last 2W.frequency:^Patient:Ord:Reported	PHQ-9
C3639361	Tension	BPRS-A
C4086709	Tension	PANSS
C3640479	Tension	HAMA

Fig. 1 Cross-lingual annotation examples of two questions of medical forms. On the left, the original English questions (OE), their German version (DE) and their translations using Google Translate (GO) are listed. On the right, the mapped UMLS concepts are shown. In UMLS, each concept is assigned with a CUI (Concept Unique Identifier). (adapted from [1])

for instance, the Epworth Sleepiness Scale (ESS) Questionnaire and the Hamilton Depression Rating Scale (HAMD). Here, an entire question is assigned to a unique corresponding concept of the ontology. In this study, we focus on such type of annotations.

Cross-lingual concept normalization denotes the process of annotating non-English documents using English concepts. This process is needed because the portion of English concepts still dominates most of the knowledge bases. For instance, one of the largest biomedical ontology sources, the Unified Medical Language System (UMLS) Metathesaurus,[2] contains more than 16.1 million terms in its current version, 2021AA. Thereof, 71% are in English, followed by 10% in Spanish and only 3% of all terms are in French or Portuguese, respectively. To augment the interoperability of non-English documents, cross-lingual concept normalization is indispensable. It is especially a necessity for finding the corresponding concepts of entire question as such concepts are not available in non-English languages.

It is common that medical forms are translated into other languages for the application in non-English speaking regions, such as for clinical or epidemiological studies. Annotating these non-English forms using the same English concepts is not only a requirement of the FDA but also enables the comparison between multiple studies carried out in different languages. Figure 1 presents examples of such cross-lingual semantic annotations. The same standardized forms in various languages shall retain conceptually equivalent meaning. Hence, many of these forms have not only been translated into a new language but also gone through some cultural adaptation and validation processes. For example, the GAD-7 (Generalized Anxiety Disorder-7) form was first published in English in 2006 [2]. It has been translated and adapted/validated into Portuguese [3], into German [4] and into Spanish [5]. The adaptation might result in further modifications on the question text, which can complicate the cross-lingual concept normalization process.

Mapping questions to concepts in the same language (normally in English) is a trivial task as the concepts are mostly syntactically identical to the question, since the concepts are derived from standardized forms. In fact, our previous studies [1, 6] show that the conventional string matching methods can already deliver good results. On the contrary, such methods perform poorly in a cross-lingual context due to text deviation caused by translation and adaptation. However, no matter cultural adaption or (machine/manual) translation, the semantics of the questions shall still be preserved. As a consequence, we proposed the idea of using deep neural network models to generate sentence embeddings as semantic representations of the questions and the concepts [1]. We achieved a substantial improvement of the annotation quality and proved that semantic embedding methods are superior to string matching based methods in a cross-lingual setting.

In this work, we expand our previous work [1] and aim to further improve the annotation quality by three means: (1) by applying new encoders (2) by injecting UMLS into new models and (3) by refining the post-processing through re-ranking annotation candidates. This study has the following main contributions:

(1) We refine the workflows of using deep network sentence encoders for cross-lingual biomedical concept normalization.
(2) We investigate the annotation quality using Biomedical Pretrained Language Models (BPLMs) as encoders.
(3) We include more state-of-the-art (SOTA) Sentence BERT (SBERT) encoders.
(4) We perform UMLS injection into the SBERT encoders and evaluate their performance.
(5) We apply candidate re-ranking using `Cross-Encoder` and test its impact on the annotation quality.
(6) We further enhance performance by combining single model results with set operations.

[2] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/release/statistics.html.

Background and Related Work

In this section, we briefly describe the recent development of the pretrained language models with the main focus on BERT (Bidirectional Encoder Representations from Transformers, [7]) and its derivatives. The BERT-models have achieved many SOTA results in various natural language processing tasks (examples see GLUE[3] and SuperGLUE[4] benchmarks). This also motivates us to integrate some of these models in our workflows for solving the concept normalization problem.

BERT consists of multi-layer bidirectional Transformer encoder based on the Transformer implementation in [8]. It was released as two sizes: $\text{BERT}_{\text{base}}$ consists of 12 Transformer layers and $\text{BERT}_{\text{large}}$ has 24 layers. BERT is trained using two unsupervised tasks: (1) masked language model (MLM) objective and (2) next sentence prediction (NSP). With the MLM, a certain percentage (usually 15%) of the input tokens are masked at random and BERT learns to predict those masked tokens. With the NSP task, BERT is trained to understand the relationship between sentences such as in Question Answering (QA) and Natural Language Inference (NLI) tasks. The initial BERT is pretrained on the BooksCorpus (800 M words) [9] and the English Wikipedia (2500 M words).

Liu et al. [10] modified the BERT's pretraining approach and proposed RoBERTa (robustly optimized BERT approach). Here, they remove the NSP objective, use dynamic masking for the MLM and increase the mini-batch size. In addition, RoBERTa is trained for more steps and with much more data (use 160 GB instead of 13 GB). These approaches have advanced BERT to a better performing model.

Since BERT is relatively resource intensive to apply, Sanh et al. [11] developed a light-weighted version of BERT, the DistilBERT. The model is compressed using the so-called *knowledge distillation* [12, 13], where a compact model—the student—is trained to reproduce the behavior of a more complex model—the teacher—by minimizing the differences between the model features. The DistilBERT comprises only 6 Transformer layers and has 40% fewer parameters. Nevertheless, it is 60% faster and still retains roughly 97% of BERT's performance on the GLUE benchmark.

MiniLM [14] is another light-weighted variant of BERT. The compression method of MiniLM, termed as *deep self-attention distillation*, is also based on knowledge distillation principles but with some modifications. The approach distills the self-attention distribution and self-attention value relation of the last Transformer layer of the teacher model. In addition, it also incorporates an intermediate-size student model, named as teacher assistant [15]. The teacher assistant distills the teacher model first and is subsequently used as the teacher to guide the training of the final student model. MiniLM outperforms DistilBERT in the majority of GLUE benchmark tasks and achieves a slightly lower average GLUE score compared to $\text{BERT}_{\text{base}}$ [14].

MPNet (*masked and permuted language modeling* [16]) was proposed to overcome two problems. The first problem is that the MLM of BERT ignores a potential dependency of the masked tokens. To address this disadvantage, XLNet [17] was introduced that uses *permuted language modeling* (PLM) as pretraining method. PLM inherits the benefits of autoregressive modeling but also allows the model to be trained in a bidirectional manner. However, it suffers from position discrepancy between pretraining and fine-tuning, which evokes the second problem. With MLM, BERT captures the position information and sees 85% of the input (if 15% of the tokens are masked). On the other hand, PLM does not have any position information, as the input sequence is presented in a permuted manner and the model only sees the preceding tokens of the to-predict token. This leads inevitably to the above-mentioned discrepancy between pretraining and fine-tuning of downstream tasks, where the model can see the entire input sequence. Consequently, MPNet introduces position compensation to PLM and alleviates the previously mentioned issues [16].

Sentence-BERT (SBERT) **and** SBERT-WK In this study, we incorporate many pretrained SBERT models [18] as our sentence encoders. Our concept normalization task involves finding the most similar pair of sentences in a large dataset. Using BERT for such type of comparison is computationally expensive as it requires each sentence pair to be input into the network separately. For a comparison of 10,000 sentences, BERT needs 50 million inference computations (~ 65 h, [18]). It is infeasible for us as the ontologies we use contain over 1 million entries. Hence, Reimers et al. [18] proposed SBERT to overcome such inefficiency. SBERT uses the above-mentioned BERT variants as backbone and adds a pooling operation (generally the mean pooling) to generate a fixed-sized sentence embedding. The models are trained using Siamese or triplet networks. The generated embeddings can be compared using similarity measures such as cosine similarity.

The SBERT-WK[5] [19] aims to refine the sentence embeddings generated by SBERT. It modifies the SBERT word embeddings based on how informative/important the word is. The importance of a word is defined by its neighboring words of the same layer and the changes of its cosine

[3] General Language Understanding Evaluation https://gluebenchmark.com.

[4] SuperGLUE https://super.gluebenchmark.com/leaderboard.

[5] WK stands for the initials of the two authors.

similarities through layers. When a word aligns well with its neighboring word vectors, it is less informative. Similarly, a word which evolves faster across layers (larger variance of the pair-wise cosine similarity), it is more important. Since this pooling strategy only alters the already generated embeddings, no further training is needed.

BERT-based biomedical pretrained language models (BPLMs) Since the publication of the BERT model in 2018 [7], various efforts have been made to adapt it for the biomedical domain. We name these as BERT-based Biomedical Pretrained Language Models (BPLMs). The earliest BPLM is BioBERT [20]. It uses the original pretrained BERT (pretrained on BooksCorpus and English Wikipedia) as base model and is further trained with PubMed abstracts and PubMed Central full-text articles (PMC). A few months later, Alsentzer et al. [21] published Clinical BERT. One of its best performing variants uses BioBERT as base model and is trained with approximately 2 million MIMIC-III v1.4 clinical notes [22]. The BlueBERT [23] can be understood as a combination of BioBERT and Clinical BERT. It has four variants depending on base model size (either $\text{BERT}_{\text{base}}$ or $\text{BERT}_{\text{large}}$) and used training corpus (trained on the PubMed corpus solely or additional with the MIMIC-III corpus). Interestingly, the large-models do not perform better than the base-models. The $\text{BERT}_{\text{base}}$-variant trained solely on the PubMed corpus is analogous to BioBERT, yet the BlueBERT variant is trained for more steps (5 M steps instead of 0.2 M steps). Experiments on various NLP tasks show that this increase in training steps does improve the results.

The above-mentioned BPLM models are all derivatives of BERT which is already pretrained on BooksCorpus and English Wikipedia. Gu et al. [24] challenge such *continual pretraining* and argue that training BERT from scratch using domain-specific corpora is more beneficial when dealing with domain-specific tasks. They pretrained the BERT model from scratch using the PubMed corpus and name their model as PubMedBERT. In addition to PubMedBERT, the authors also create a new benchmark, the Biomedical Language Understanding and Reasoning Benchmark (BLURB), which comprises biomedical NLP tasks focusing on PubMed-based applications. PubMedBERT outperforms the above-mentioned models in almost every BLURB task (only BioBERT is better in 2 of the 13 tasks). Hence, they conclude that to solve domain-specific tasks, it is better to use models entirely pretrained on domain-specific corpora than to use models that have already been trained with out-domain corpora.

UMLS injected BPLMs Various studies have shown that for named-entity recognition or concept normalization tasks extra training of the language models on a given knowledge base is beneficial [16, 25–28]. Thus, we incorporate two such models in our workflows: CODER [27] and SapBERT [26]. Both approaches propose pretraining using UMLS synonyms, referred to as UMLS injection. In addition, CODER also embeds the relationships between the concepts into the vector representation. CODER has two versions. The English version, $\text{CODER}_{\text{ENG}}$, uses PubMedBERT as base model and the multilingual version, $\text{CODER}_{\text{ALL}}$, uses multilingual BERT as base model. Both versions differ also in training corpus: $\text{CODER}_{\text{ENG}}$ utilizes only the English concepts in the UMLS while $\text{CODER}_{\text{ALL}}$ is trained on concepts of all languages. CODER applies contrastive learning. Here, term representations are learnt by maximizing cosine similarity between positive term-term pairs (i.e., between synonyms of a given concept) and term-relation-term pairs.

The SapBERT achieves many SOTA results on the medical entity linking (MEL) benchmark. The *Self-Alignment Pretraining* (SAP) is a procedure that learns to self-align synonyms in the UMLS and can also be used for fine-tuning on task-specific datasets. During pretraining, an online hard triplet mining is necessary to locate the most informative training examples. With each mini-batch, all possible triplets for all terms are constructed. A triplet (x_a, x_p, x_n) contains an anchor x_a, an arbitrary term in the mini-batch and the x_p and x_n denote each either a positive or a negative match of the x_a. Only the triplets are retained for pretraining if they satisfy the following constraint :

$$\|f(x_a) - f(x_p)\|_2 < \|f(x_a) - f(x_n)\|_2 + \lambda$$

where f is modeled by a BERT model and λ is a predefined margin. In other words, only triplets with negative samples that are very similar (in the paper they use cosine similarity) to the positive sample by a margin of λ are kept for pretraining. They use Multi-Similarity loss function [29] as learning objective that leverages the similarities among and between positive and negative pairs.

Multilingual pretrained language models In the current study we apply several multilingual pretrained language models. We choose the pretrained models developed by the same authors of SBERT. Further, we also include the multilingual versions of CODER (described previously) and SapBERT in our workflows.

Reimers et al. [30] proposed *multilingual knowledge distillation* that seeks to reinforce better alignment of the multilingual sentence embeddings, i.e., the sentence embeddings of different languages shall be mapped to the same vector space if they are semantically equivalent. Through the distillation, the student model $\hat{M}$, generally (but not restricted to) a smaller multilingual pretrained model, learns the behavior of the teacher model M, generally an intensively trained monolingual (English) model. The pretraining requires a set of parallel (translated) sentences $((s_1, t_1), \ldots, (s_n, t_n))$ where t_i is the translation of s_i. The learning objective is to minimize the mean squared loss so that $\hat{M}(s_i) \approx M(s_i)$ and $\hat{M}(t_i) \approx M(s_i)$.

The multilingual version of SapBERT, later referred to as SapBERT-XLMR, differs from the English version in two folds. Firstly, it is trained with UMLS terms of all languages. Secondly, the pretraining also incorporates general-domain translation data, including "muse" word translations [31] and parallel Wikipedia article titles. The original and the translated sequences are considered as synonyms for the SAP training process.

Methods

Corpus and Ontology

This study uses the same 21 German medical forms and the 497 questions as in [1]. Many of the forms are utilized in the LIFE[6] Adult Study [32], a large scale cohort study investigating the factors leading to civilization diseases, such as vascular disease, heart function, allergies and depression. Examples of the included medical forms are the Patient Health Questionnaire (PHQ, [33]) and the GAD-7.

The UMLS Metathesaurus is one of the largest biomedical ontology sources by far. We consequently choose UMLS so that we can maximize the semantic interoperability for our corpus. Since some of the pretrained models that are applied in this study (namely CODER and SapBERT) use the UMLS version 2020AA for concept injection, we also limited ourselves to the same version for our annotation task for a fair comparison. The UMLS version 2020AA integrates 214 source vocabularies and contains approximately 4.28 million concepts. To improve annotation efficiency and since not all ontologies in the UMLS are relevant, we selected three source ontologies from the UMLS that still cover 99.1% of the GSC annotations [1]. The selected subset contains all concepts from (1) the NCI Thesaurus, (2) the LOINC, and (3) the Consumer Health Vocabulary. In total, the subset includes 1,115,090 terms belonging to 399,758 concepts.

In order to evaluate the annotation quality, we manually annotated the medical forms using the selected UMLS subset and built a Gold Standard Corpus (GSC) [1]. Overall, we identified 1105 GSC annotations. Their frequency distribution of number of annotations per question is shown in Fig. 2. In the GSC, most of the questions have up to 2 annotations and about 10% of the questions have 3 or 4 annotations. There are only a few questions being mapped to more than 5 UMLS concepts. From our observations, the duplication is mainly due to (1) same question of a form might be given multiple CUIs in the UMLS or (2) the same question occurs in different forms and hence has different CUIs. Figure 1 shows such examples.

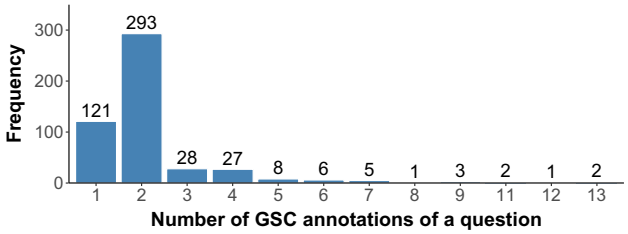


Fig. 2 Frequency distribution of number of annotations of a question in the GSC (adapted from [1])

Annotation Workflows

We design two workflows: (1) *Workflow-Multi* and (2) *Workflow-MT* to tackle the cross-lingual concept normalization problem (Fig. 3). In *Workflow-Multi* we input the German forms directly into a given multilingual sentence encoder to generate sentence embeddings. We use the same encoder to encode the embeddings for the English concepts in the UMLS (Fig. 3a). In *Workflow-MT* (MT stands for Machine Translation), we first translate the German forms into English using three machine translators (DeepL,[7] Microsoft Translator[8] and Google Translate[9]) (Fig. 3b). We then generate the embeddings of the translated questions and the English UMLS concepts using a given sentence encoder. The sentence encoders we used in *Workflow-MT* are not limited to English encoders but also include multilingual ones. In a preliminary study we observed that multilingual encoders we selected to generate English sentence embeddings can also achieve good annotation quality. After the encoding process, cosine similarity is computed between each pair of question and a candidate concept embeddings. These mappings are ranked and the Top k results are retained for evaluation, where $k \in \{1, 2, 3, 5\}$. We apply the metrics precision, recall and F-measure to evaluate our results. We also use *Workflow-MT* to annotate the original English corpus for the reference comparison.

There are four optional components in the *Workflow-MT*, which are presented in dashed lines in Fig. 3. First, the UMLS injection indicates that we train the sentence encoders using concepts in the UMLS to refine the sentence encoders. The methods and encoders used for training are detailed in "UMLS Injected SBERTv2 Models ($MG_{SapFull}$ and $MG_{SapSubset}$)". Second, we incorporate the SBERT-WK

[6] LIFE stands for Leipzig Research Center for Civilization Diseases https://life.uni-leipzig.de/en/life_health_study.html.

[7] https://www.deepl.com/translator.

[8] https://www.microsoft.com/en-us/translator/.

[9] https://translate.google.com.

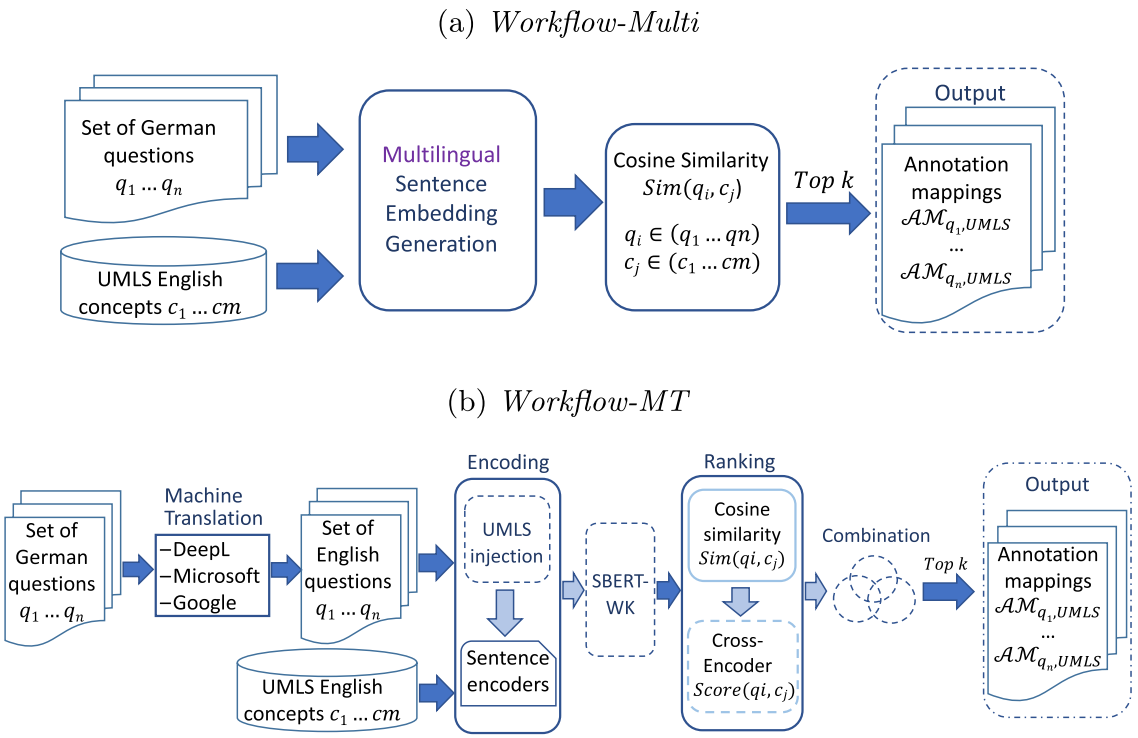


Fig. 3 Two workflows to generate cross-lingual annotations using sentence encoders

[19] as we observed that applying SBERT-WK to the English embeddings generated by SBERT models does improve the annotation quality significantly [1]. The third and forth optional components in the *Workflow-MT* are extra post-processing steps. The `Cross-Encoder` is used to rerank the candidates. In the combination step, set operations are applied to the result sets generated by different translated corpora. See "Post-processing" for more details about these post-processing methods.

Baseline: AnnoMap

AnnoMap [34, 35] is a conventional string matcher that generates candidates using three string similarity functions: TF/IDF, Trigram and LCS (longest common substring). After candidate generation, an optional group-based selection can be applied to improve precision. AnnoMap retains candidates whose similarity scores are above a given threshold δ. We set two thresholds $\delta \in \{0.6, 0.7\}$ that generally generate the best F-measures. We also retain the same result sizes k as in workflows using language pretrained models, i.e., $k \in \{1, 2, 3, 5\}$. To be able to obtain the desired result set sizes, we did not apply group-based selection in this study because it might return fewer candidates than a given k. We annotate the same three translated corpora as in the *Workflow-MT* and also the original English corpus as reference.

Model-Groups

In total, we applied 53 English and multilingual BERT-based pretrained language models that are grouped into five Model-Groups: $MG_{SBERTv1}$, MG_{BPLM}, $MG_{SBERTv2}$, $MG_{SapFull}$ and $MG_{SapSubset}$. First group, $MG_{SBERTv1}$, includes the ten English SBERT models we used in our previous study [1] as reference. These models are selected from SentenceTransformers-v1 and are listed in Table 1. Be noted that since we use different UMLS versions in both studies (2019AB

Table 1 Selected models from SentenceTransformers-v1 ($MG_{SBERTv1}$)

Model name	Base model	Size	Training
$BERT_{base}$-N	BERT	Base	NLI
$BERT_{base}$-NS	BERT	Base	NLI + STSb
$BERT_{large}$-N	BERT	Large	NLI
$BERT_{large}$-NS	BERT	Large	NLI + STSb
$RoBERTa_{base}$-N	RoBERTa	Base	NLI
$RoBERTa_{base}$-NS	RoBERTa	Base	NLI + STSb
$RoBERTa_{large}$-N	RoBERTa	Large	NLI
$RoBERTa_{large}$-NS	RoBERTa	Large	NLI + STSb
DistilBERT-N	DistilBERT	6 layers	NLI
DistilBERT-NS	DistilBERT	6 layers	NLI + STSb

Mean pooling was applied in the SBERT training phase of all models

Table 2 Selected English and multilingual BPLM models

Model name	Base model	Representation	Pretraining	UMLS injection
(a) English encoders				
BioBERT	BERT (Wikipedia, BookCorpus)	[CLS]	PubMed	–
$\text{PubMedBERT}_{\text{abstract}}$	BERT from scratch	[CLS]	PubMed	–
$\text{PubMedBERT}_{\text{full}}$	BERT from scratch	[CLS]	PubMed, PMC	–
$\text{SapBERT}_{\text{CLS}}$	$\text{PubMedBERT}_{\text{full}}$	[CLS]	PubMed, PMC	2020AA English
$\text{SapBERT}_{\text{mean}}$	$\text{PubMedBERT}_{\text{full}}$	Mean-token	PubMed, PMC	2020AA English
$\text{CODER}_{\text{ENG}}$	$\text{PubMedBERT}_{\text{abstract}}$	[CLS]	PubMed	2020AA English
(b) Multilingual encoders				
$\text{SapBERT-XLMR}_{\text{base}}$	$\text{XLMR}_{\text{base}}$	[CLS]	Multilingual CommonCrawl	2020AB multilingual
$\text{SapBERT-XLMR}_{\text{large}}$	$\text{XLMR}_{\text{large}}$	[CLS]	Multilingual CommonCrawl	2020AB multilingual
$\text{CODER}_{\text{ALL}}$	mBERT	[CLS]	Multilingual Wikipedia	2020AA multilingual

All models consists of 12 transformer layers as $\text{BERT}_{\text{base}}$ except $\text{SapBERT-XLMR}_{\text{large}}$ is based on $\text{BERT}_{\text{large}}$

PubMed PubMed abstracts, *PMC* PubMed Central full-text articles

in [1] and 2020AA in current work), the results differ. The description of other Model-Groups are detailed as follows.

Biomedical Pretrained Language Models (MG_{BPLM})

We select nine BERT-based BPLMs from BioBERT, PubMedBERT, SapBERT and CODER (Table 2). Six of them are English encoders, which are trained using English text and the other three are multilingual encoders. PubMedBERT differs from BioBERT in that it is trained from scratch using PubMed corpus while BioBERT was also pretrained with English Wikipedia and BookCorpus. There are two versions of PubMedBERT, one version is pretrained only with abstracts of PubMed (named $\text{PubMedBERT}_{\text{abstract}}$) whereas the other one is also trained on full text of PubMed articles ($\text{PubMedBERT}_{\text{full}}$). Both SapBERT and CODER use PubMedBERT as base-model and are injected with the UMLS 2020AA full version. Pretrained SapBERT models are available in two versions: one with [CLS] representation ($\text{SapBERT}_{\text{CLS}}$) and one with mean-token ($\text{SapBERT}_{\text{mean}}$). We applied both the English and the multilingual encoders in MG_{BPLM} for the *Workflow-MT* and eventually only the multilingual encoders for *Workflow-Multi*.

SBERTv2 Models ($\text{MG}_{\text{SBERTv2}}$)

Since the publication of our previous study [1], various new models are trained into Sentence-BERT and are included in the SentenceTransformers-v2.[10] Among them, we selected ten English models and four multilingual models that have shown to yield good results in NLP tasks and vary in efficiency (Table 3).

English encoders The selected ten English SBERTv2 models are mainly derived from MPNet, RoBERTa, DistilRoBERTa and MiniLM. They are fine-tuned on three different training sets: NLI + STSb, paraphrase and ALL. The NLI + STSb training set includes the NLI datasets[11] and the STSb dataset [38]. Additional corpora[12] are used to train the models using the paraphrase training set. The ALL corpus further expands the paraphrase training set into a dataset including one billion sentence pairs from various sources.[13] We also apply the optional SBERT-WK to the new SBERTv2 English encoders.

Multilingual encoders We choose three new multilingual models from the SentenceTransformers-v2, which are listed as M2-M4 in Table 3. We retain the best performing model in our previous study (M1) for comparison. Unlike the models in MG_{BPLM}, we apply only the English encoders in $\text{MG}_{\text{SBERTv2}}$ for *Workflow-MT* as, according to our preliminary study, the SBERTv2 multilingual encoders do not generate good annotation results using *Workflow-MT*.

UMLS Injected SBERTv2 Models ($\text{MG}_{\text{SapFull}}$ and $\text{MG}_{\text{SapSubset}}$)

Among the BPLM models, CODER and SapBERT are both with UMLS injection. Based on our research results,

[10] https://www.sbert.net/index.html.

[11] Containing the Stanford Natural Language Inference dataset [36] and the Multi-Genre NLI dataset [37].

[12] Sentence-compression, SimpleWiki, altlex, msmarco-triplets, quora-duplicates, coco-captions, flickr30k-captions, yahoo-answers-title-question, S2ORC-citation-pairs, stackexchange-duplicate-questions, wiki-atomic-edits.

[13] https://huggingface.co/sentence-transformers/all-mpnet-base-v2.

Table 3 Selected models from SentenceTransformers-v2

(a) English encoders

Model name	Base model	Size	Training
MPNet-STSb	MPNet	Base	NLI + STSb
RoBERTa-STSb	RoBERTa	Base	NLI + STSb
DistilRoBERTa-STSb	DistilRoBERTa	6 layers	NLI + STSb
MPNet-Paraphrase	MPNet	Base	Paraphrase
MiniLM(L12)-Paraphrase	MiniLM	Base	Paraphrase
MiniLM(L6)-Paraphrase	MiniLM	6 layers	Paraphrase
MPNet-ALL	MPNet	Base	All
DistilRoBERTa-ALL	DistilRoBERTa	Base	All
MiniLM(L12)-ALL	MiniLM	Base	All
MiniLM(L6)-ALL	MiniLM	6 layers	All

(b) Multilingual encoders

Model code	Teacher model	Student model	Languages
M1	mUSE	DistilmBERT	15 languages
M2	mUSE	DistilmBERT	50+ languages
M3	MPNet-Paraphrase	XLM-R	50+ languages
M4	MiniLM(L12)-Paraphrase	XLM-R	50+ languages

Mean pooling was applied in the SBERT training phase of all models

SapBERT models perform significantly better than CODER (see Table 8). Consequently, we use the method proposed in SapBERT [26][14] to inject UMLS 2020AA into the English models of $MG_{SBERTv2}$. For the injection, we use either the full version of the UMLS ($MG_{SapFull}$) or the selected subset of the UMLS ($MG_{SapSubset}$, subset selection see "Corpus and Ontology"). Since these UMLS injected models are SBERT-based, we are also able to apply SBERT-WK to them.

In the *Workflow-MT*, a configuration, denoted as `config` in the following text, is determined by a given model, with or without SBERT-WK, different translated corpora and the various result sizes. Table 4 shows the number of models of each Model-Group and the corresponding number of `configs` used to generate annotation results.

Post-processing

Combination using set operations Our previous studies [1, 6, 39] show that combining annotation results using set operations can further improve annotation quality. We also conclude that combining result sets of the three different translated corpora deliver the best quality [1]. Hence, within each Model-Group, we combine the annotation candidates generated by the single `configs` of the three different translated corpora (Google Translate, DeepL and Microsoft Translator) using intersection, union and 2-vote-agreement (an annotation is considered as correct by at least two of the three `configs`) in this study.

Cross-Encoders For finding the most similar sentence pair, if two sentences are passed into the encoder-network simultaneously, such a network is named `Cross-Encoder` [40]. Thakur et al. [41] show that a fine-tuned `Cross-Encoder` (BERT) delivers better results for the STS Benchmark than a fine-tuned Bi-Encoder (SBERT). However, since the sentences are passed to the network in pairs, using `Cross-Encoder` for finding most similar sentences is computationally expensive as it demands quadratic time complexity. To overcome this inefficiency and still taking the advantage of the better result quality of the `Cross-Encoder`, we apply the `Cross-Encoder` on a limited

Table 4 Number of models and configurations used in *Workflow-MT*

Model-Group	Models	SBERT-WK	Corpus	Result size	`config`
$MG_{SBERTv1}$	10	With or w/o	GO DL MS	$k \in \{1, 2, 3, 5\}$	192
MG_{BPLM}	9	n.a.			108
$MG_{SBERTv2}$	10	With or w/o			160
$MG_{SapFull}$	10				160
$MG_{SapSubset}$	10				160

In column SBERT-WK, "n.a." indicates that SBERT-WK is not applicable to BPLM models

GO Google Translate, *DL* DeepL, *MS* Microsoft Translator

[14] https://github.com/cambridgeltl/sapbert.

Table 5 The best precision, recall and F-measure obtained from AnnoMap with thresholds $\delta \in \{0.6, 0.7\}$ and of result sizes $k \in \{1, 2, 3, 5\}$

Corpus	δ	Result size	Precision	Recall	F-measure
Precision					
GO	0.7	Top1	61.60	13.94	22.73
OE	0.7	Top1	93.60	41.00	57.02
Recall					
GO	0.6	Top5	35.50	36.11	35.80
OE	0.6	Top5	50.91	86.15	64.00
F-measure					
GO	0.6	Top2	53.04	30.05	38.36
OE	0.6	Top2	86.32	73.67	79.49

The last row of each metric in grey is the best corresponding result obtained using the original English corpus (OE)

GO Google Translate

candidate list. We first reduce the search space by generating a short list of candidates using the standard cosine similarity ranking. We then use the `Cross-Encoder` to rerank these candidates and evaluate the Top*k* results accordingly. We utilize the implementation of `Cross-Encoder` in the SentenceTransformers-v2.[15] We select the `Cross-Encoder` model *stsb-roberta-large* as it delivers the best results for the STS benchmark. The `Cross-Encoder` returns a score for each given sentence pair. Given an encoder `config`, a sentence pair comprises a question of the given translated corpus (the reference sentence) and one of the candidates found by that `config`. For each single `config`, we retain the best 50 candidates for reranking. For each combination, we first rerank the 50 candidates generated by each single `config` and then apply the set operations to combine the reranked candidates to obtain the final annotation results.

Evaluation

In this section, we first present the annotation quality of AnnoMap, the conventional string matching method. We then report the results of the proposed workflows: *Workflow-Multi* and *Workflow-MT*. In addition to annotation quality, we also investigate the computation efficiency of the models used in *Workflow-MT* and the combination results. Further, we give a reflection on the relationship between recall and result size. At the end of the section, we summarize our main findings.

[15] https://www.sbert.net/examples/applications/cross-encoder/README.html.

Table 6 Averaged annotation quality using *Workflow-Multi*

Model name	$M_{\text{Precision}}$	M_{Recall}	$M_{\text{F-measure}}$
MG_{BPLM}			
$\text{SapBERT-XLMR}_{\text{large}}$	48.22	50.59	45.85
$\text{SapBERT-XLMR}_{\text{base}}$	40.48	42.15	38.34
$\text{CODER}_{\text{ALL}}$	14.48	16.13	14.13
$\text{MG}_{\text{SBERTv2}}$			
M3	44.18	46.76	42.11
M1	43.88	45.95	41.63
M2	41.40	43.46	39.32
M4	37.55	39.30	35.62

For the model names refer to Table 2 and Table 3. Models are ranked by F-measure within each Model-Group

Baseline: AnnoMap

The best results of the conventional string matching method, AnnoMap, are shown in Table 5. When using the original English corpus, AnnoMap obtain the best precision of 93.6%, best recall of 86% and best F-measure of 79.49%. However, when annotating translated corpora, the AnnoMap performs far less well: 32% reduction in precision, 50.4% in recall and 41.13% in F-measure. The large drop in recall indicates that the paraphrase of the questions after translation/cultural adaptation prevents the conventional string matching method from finding the correct annotations, especially in recall. The results of AnnoMap also show that the most suitable machine translator is Google Translate while the annotation quality using DeepL and Microsoft Translator are worse.

Annotation Using Pretrained Language Models

Workflow-Multi

The *Workflow-Multi* has the advantage of not requiring machine translators but uses the German questions as input. We applied the three multilingual encoders in the MG_{BPLM} (Table 2) and the four multilingual encoders in the $\text{MG}_{\text{SBERTv2}}$ (Table 3) for this workflow. Table 6 presents the averaged annotation quality of these multilingual models. Among the MG_{BPLM} models, SapBERT models perform significantly better than the $\text{CODER}_{\text{ALL}}$. Actually, $\text{CODER}_{\text{ALL}}$ is the worst performing multilingual model among all. Among the $\text{MG}_{\text{SBERTv2}}$ models, M3 performs best and is better than the best model we tested in our previous study [1]. The best multilingual encoder is the $\text{SapBERT-XLMR}_{\text{large}}$. It gains approximately 8% more than its base model ($\text{SapBERT-XLMR}_{\text{base}}$) in every averaged metric and also outperforms the best $\text{MG}_{\text{SBERTv2}}$ multilingual model (M3). When comparing results of single `configs`,

Table 7 Best performing `config` for each metric using *Workflow-Multi* in MG_{BPLM} and $MG_{SBERTv2}$

Metric	Model name	Result size	Precision	Recall	F-measure
MG_{BPLM}					
Precision	SapBERT-XLMR$_{large}$	1	**60.16**	27.06	37.33
Recall	SapBERT-XLMR$_{large}$	5	31.74	**68.69**	43.42
F-measure	SapBERT-XLMR$_{large}$	2	56.44	48.78	**52.33**
$MG_{SBERTv2}$					
Precision	M1	1	**56.74**	25.52	35.21
Recall	M3	5	28.98	**63.62**	39.82
F-measure	M1	2	51.69	45.70	**48.51**

Table 8 Averaged annotation quality of BPLM models

Model name	$M_{Precision}$	M_{Recall}	$M_{F\text{-}measure}$
SapBERT-XLMR$_{large}$	51.94	54.22	49.22
SapBERT-XLMR$_{base}$	49.28	51.48	46.74
SapBERT$_{mean}$	43.80	45.85	41.55
SapBERT$_{CLS}$	43.16	45.17	40.98
CODER$_{ALL}$	39.92	42.35	38.10
CODER$_{ENG}$	39.10	41.85	37.49
PubMedBERT$_{abstract}$	25.51	27.66	24.59
PubMedBERT$_{full}$	20.22	21.63	19.36
BioBERT	15.99	17.34	15.43

For the model details refer to Table 2. Models are ranked by F-measure

the superiority of the model over other multilingual encoders can be seen again (Table 7). Using *Workflow-Multi*, SapBERT-XLMR$_{large}$ generates 60.16% as the best precision, 68.69% as best recall and 52.33% as best F-measure.

Workflow-MT

The following presents the results of *Workflow-MT*. We test if integrating machine translators into cross-lingual biomedical concept normalization workflow improves the annotation quality. For this workflow, we also use multilingual encoders in MG_{BPLM} to encode English corpora (original English and the three translated English corpora).

BPLM models Table 8 presents the performance of the nine models in the MG_{BPLM}, including 6 English encoders and 3 multilingual encoders. Interestingly, the best two models are the multilingual models (SapBERT-XLMR$_{large}$ and SapBERT-XLMR$_{base}$). The models with UMLS injection (SapBERT and CODER models) outperform the models without UMLS injection (PubMedBERT and BioBERT) remarkably. All SapBERT models exceed CODER models. Notably, the two multilingual SapBERT models (SapBERT-XLMR$_{large}$ and SapBERT-XLMR$_{base}$) achieve better results than the two English SapBERT models (SapBERT$_{mean}$ and SapBERT$_{CLS}$). Similarly, the multilingual CODER (CODER$_{ALL}$) is also better than the English CODER (CODER$_{ENG}$).

When comparing the averaged results of the same multilingual models but using different workflows (Table 6 and Table 8), we observe that the multilingual models deliver better results using *Workflow-MT* than *Workflow-Multi*. The SapBERT-XLMR$_{large}$ improves 3.37% in F-measure using *Workflow-MT* (from 45.85 to 49.22%) and the SapBERT-XLMR$_{base}$ gains an even larger increase of 8.38% in F-measure (from 38.34 to 46.72%). The CODER$_{ALL}$ performs dramatically different on using different workflows, in *Workflow-Multi* it only reaches an averaged F-measure of 14.13% while using *Workflow-MT* it achieves an averaged F-measure of 38.10%.

UMLS injection of SBERTv2 models Table 9 presents the averaged annotation quality of models without UMLS injection ($MG_{SBERTv2}$), those injected with 2020AA UMLS full version ($MG_{SapFull}$) and those injected with the selected subset ($MG_{SapSubset}$). UMLS injection is beneficial for 8 of the 10 models (except MiniLM(L12)-ALL and MiniLM(L6)-ALL). We also observe that UMLS injection improves different models in various magnitudes. The most significant improvement is seen by DistilRoBERTa-ALL. Before UMLS injection, RoBERTa-STSb delivers the best averaged annotation quality (colored in blue). After UMLS injection (with either full version or subset), DistilRoBERTa-ALL becomes the best model. The second best model, MPNet-ALL, can also outperform RoBERTa-STSb after UMLS injection.

We conduct pairwise *t*-test to compare the annotation metrics of the same model between different Model-Groups to test the effect of UMLS injection statistically. Each comparison is done between the identical `configs` of the same model between two Model-Groups: $MG_{SBERTv2}$ against $MG_{SapFull}$, $MG_{SBERTv2}$ against $MG_{SapSubset}$ and $MG_{SapFull}$ against $MG_{SapSubset}$. The results are shown as superscripts in Table 9. Only one model, MiniLM(L6)-ALL, performs better without UMLS injection statistically (p-value $<$ 0.01). Among the eight models that benefit from the UMLS injection, five (of $MG_{SapFull}$) and six (of $MG_{SapSubset}$) of them perform significantly better than uninjected models (denoted with ** in the table). When comparing the results

Table 9 Averaged annotation quality of $MG_{SBERTv2}$, $MG_{SapFull}$ and $MG_{SapSubset}$ models

Model name	$M_{Precision}$	M_{Recall}	$M_{F\text{-}measure}$
$MG_{SBERTv2}$			
MiniLM(L12)-ALL	49.74	51.72	47.08
MiniLM(L6)-ALL	48.88[a]	50.90[a]	46.30[a]
DistilRoBERTa-ALL	48.77	50.88	46.21
MPNet-ALL	49.47	51.64	46.90
MiniLM(L12)-Paraphrase	45.63	47.97	43.37
MiniLM(L6)-Paraphrase	44.20	46.59	42.09
MPNet-Paraphrase	50.26	52.10[c]	47.50
DistilRoBERTa-STSb	48.52	51.14	46.15
MPNet-STSb	50.48	52.62	47.80[a]
RoBERTa-STSb	50.58	53.22	48.08
$MG_{SapFull}$			
MiniLM(L12)-ALL	47.80	49.71	45.24
MiniLM(L6)-ALL	45.48	47.48	43.12
DistilRoBERTa-ALL	52.65[b]	54.34[b]	49.65[b]
MPNet-ALL	52.28[b]	54.09[b]	49.40[b]
MiniLM(L12)-Paraphrase	48.47[b]	50.44[b]	45.87[b]
MiniLM(L6)-Paraphrase	44.39	46.46	42.10
MPNet-Paraphrase	**50.78**	**52.48**	**47.95**
DistilRoBERTa-STSb	50.81[b]	53.19[b]	48.19[b]
MPNet-STSb	50.58	52.27	47.76
RoBERTa-STSb	52.00[b]	54.21[b]	49.24[b]
$MG_{SapSubset}$			
MiniLM(L12)-ALL	47.79	49.72	45.24
MiniLM(L6)-ALL	45.43	47.43	43.08
DistilRoBERTa-ALL	52.58[b]	54.34[b]	49.61[b]
MPNet-ALL	52.24[b]	54.04[b]	49.36[b]
MiniLM(L12)-Paraphrase	**48.63**[b]	**50.62**[b]	**46.03**[b]
MiniLM(L6)-Paraphrase	**45.59**[b]	**47.82**[b]	**43.32**[b]
MPNet-Paraphrase	50.19	51.49	47.25
DistilRoBERTa-STSb	49.98[b]	52.32[b]	47.39[b]
MPNet-STSb	50.80	52.41	47.92
RoBERTa-STSb	51.95[b]	54.11[b]	49.17[b]

For the model details refer to Table 3. The best models of each Model-Group are in blue. The [a] indicates the models of $MG_{SBERTv2}$ are significantly better than models in both $MG_{SapFull}$ and $MG_{SapSubset}$ using pairwise *t*-tests (*p*-value < 0.01). Contrastingly, [b] shows the models of $MG_{SapFull}$ and $MG_{SapSubset}$ are statistically better than those of $MG_{SBERTv2}$. [c] specifies the models of $MG_{SBERTv2}$ are better than those of $MG_{SapSubset}$. The better models are in bold when comparing the metrics between $MG_{SapFull}$ and $MG_{SapSubset}$

between injection using full version ($MG_{SapFull}$) or selected subset ($MG_{SapSubset}$), the differences are mostly insignificant: merely two models are better using selected subset (MiniLM(L12)-Paraphrase and MiniLM(L6)-Paraphrase, in bold), while one model (MPNet-Paraphrase) is better using full version. Hence, we can conclude that UMLS injection into SBERTv2 models is generally beneficial for our biomedical concept normalization task, though various effectiveness is observed. Moreover, injection using a relevant subset is sufficient and also more efficient than injecting the full version of the UMLS.

Best `configs` of *Workflow-MT* The best performing single `configs` in precision, recall and F-measure are shown in Tables 10, 11, 12, respectively. We present the best 5 results within each Model-Group. The last row in gray of each model group is the best results using original English corpus as an indication of upper bound. The first three metric columns are the results using standard workflow, i.e., the candidates are ranked using the cosine similarities of the mappings. The last three metric columns show the results that are reranked using `Cross-Encoder`. Overall, we can exceed our previous results in [1] (comparable results shown as $MG_{SBERTv1}$ models in Tables 10, 11, 12) in all metrics. The best annotation quality generated by *Workflow-MT* on annotating original English corpus exceeds that of conventional string matching. We can push the upper limit for a further 6.04% in recall (92.04 vs 86%) and 2% in the best precision and F-measure (precision: 95.98 vs. 93.6%, F-measure: 81.34 vs 79.49%).

We achieve the best precision of 71.23% with both standard ranking and reranking with `Cross-Encoder` (Table 10). Reranking using `Cross-Encoder` can improve the precision results for almost all the 25 `configs` (only 4 cases in $MG_{SapFull}$ and $MG_{SapSubset}$ are exceptions). The best recall of standard ranking is 73.67% by the best BPLM model (SapBERT-$XLMR_{large}$ with Google Translate corpus, Table 11). Using `Cross-Encoder` for reranking, we can improve the best recall further to 74.84% with RoBERTa-STSb using Google Translate and with SBERT-WK. Actually, this `config` delivers the best recalls within each of the SBERTv2 Model-Groups. Similar to the best precision results, reranking using `Cross-Encoder` improves most best recall results except from three `configs` (two `configs` in MG_{BPLM} and one `config` in $MG_{SBERTv2}$). The best F-measure, 61.90%, is delivered by DistilRoBERTa-ALL using Google Translate with SBERT-WK in $MG_{SapSubset}$ (Table 12). On the other hand, the best F-measure using `Cross-Encoder` does not exceed this result. But in general, reranking using `Cross-Encoder` is also beneficial for F-measure results. The F-measures of only 4 `configs` (two in $MG_{SapFull}$ and two in $MG_{SapSubset}$) are not improved by reranking.

The best performing model in MG_{BPLM} is the SapBERT-$XLMR_{large}$. It achieves the best 3 results in every metric using the 3 different translated corpora in the order of Google Translate, DeepL and Microsoft Translator (Tables 10, 11, 12). An interesting observation is, since the model is a multilingual encoder, it is also applied in the *Workflow-Multi*. Comparing its best results using *Workflow-Multi* (Table 7) and those using *Workflow-MT*, using translated corpora can deliver even

Table 10 Best single `configs` in precision within each Model-Group using only cosine similarity for ranking or additionally reranked with `Cross-Encoder`

Model	SBERT-WK	Corpus	Result size	Standard			`Cross-Encoder`		
				P	R	F	P	R	F
$MG_{SBERTv1}$									
DistilBERT-NS	With	GO	Top1	66.80	30.05	41.45	70.62	31.76	43.82
$BERT_{large}$-NS	W/o	GO	Top1	66.60	29.95	41.32	70.62	31.76	43.82
$RoBERTa_{base}$-N	With	GO	Top1	65.79	29.59	40.82	70.02	31.49	43.45
$BERT_{base}$-N	With	GO	Top1	65.59	29.50	40.70	70.02	31.49	43.45
$RoBERTa_{large}$-NS	W/o	GO	Top1	65.39	29.41	40.57	69.82	31.40	43.32
$BERT_{large}$-NS	W/o	OE	Top1	94.77	42.62	58.80	92.35	41.54	57.30
MG_{BPLM}									
SapBERT-$XLMR_{large}$	n.a.	GO	Top1	67.61	30.41	41.95	69.22	31.13	42.95
SapBERT-$XLMR_{large}$	n.a.	DL	Top1	66.40	29.86	41.20	70.02	31.49	43.45
SapBERT-$XLMR_{large}$	n.a.	MS	Top1	63.98	28.78	39.70	69.01	31.04	42.82
SapBERT-$XLMR_{base}$	n.a.	GO	Top1	63.98	28.78	39.70	65.19	29.32	40.45
SapBERT-$XLMR_{large}$	n.a.	GO	Top2	62.75	54.57	58.37	63.38	57.01	60.03
SapBERT-$XLMR_{base}$	n.a.	OE	Top1	94.57	42.53	58.68	91.95	41.36	57.05
$MG_{SBERTv2}$									
RoBERTa-STSb	With	GO	Top1	67.81	30.50	42.07	70.22	31.58	43.57
MPNet-Paraphrase	With	DL	Top1	66.60	29.95	41.32	70.62	31.76	43.82
MPNet-STSb	With	DL	Top1	66.40	29.86	41.20	70.22	31.58	43.57
MiniLM(L12)-ALL	With	DL	Top1	66.40	29.86	41.20	**71.23**	32.04	44.19
MPNet-STSb	With	GO	Top1	66.00	29.68	40.95	69.42	31.22	43.07
MiniLM(L12)-ALL	With	OE	Top1	95.17	42.81	59.05	92.76	41.72	57.55
$MG_{SapFull}$									
DistilRoBERTa-ALL	With	MS	Top1	70.42	31.67	43.70	67.00	30.14	41.57
DistilRoBERTa-ALL	With	GO	Top1	70.22	31.58	43.57	69.22	31.13	42.95
DistilRoBERTa-ALL	With	DL	Top1	68.81	30.95	42.70	69.82	31.40	43.32
RoBERTa-STSb	With	GO	Top1	68.61	30.86	42.57	70.02	31.49	43.45
MPNet-ALL	With	GO	Top1	68.21	30.68	42.32	69.22	31.13	42.95
MiniLM(L6)-Paraphrase	W/o	OE	Top1	**95.98**	43.17	59.55	**93.16**	41.90	57.80
$MG_{SapSubset}$									
DistilRoBERTa-ALL	With	GO	Top1	**71.23**	32.04	44.19	69.62	31.31	43.20
RoBERTa-STSb	With	GO	Top1	69.22	31.13	42.95	70.42	31.67	43.70
DistilRoBERTa-ALL	With	MS	Top1	69.22	31.13	42.95	66.00	29.68	40.95
DistilRoBERTa-ALL	With	DL	Top1	69.01	31.04	42.82	69.82	31.40	43.32
MPNet-ALL	With	GO	Top1	68.01	30.59	42.20	69.01	31.04	42.82
DistilRoBERTa-ALL	W/o	OE	Top1	**95.98**	43.17	59.55	91.95	41.36	57.05

In column SBERT-WK, "n.a." indicates that SBERT-WK is not applicable to BPLM models. The last row of each metric in gray is the best corresponding result obtained using the original English corpus (OE). The best precision of all `configs` of translated corpora and OE are in bold

GO Google Translate, *DL* DeepL, *MS* Microsoft Translator, *P* precision, *R* recall, *F* F-measure

better results. When inputing the German forms directly into SapBERT-$XLMR_{large}$, the best precision, recall and F-measure are 60.16, 68.69 and 52.33%. On the other hand, using Google Translate translated corpus as input, it achieves 67.61% in precision, 73.69% in recall and 58.37% in F-measure. This implies that the alignment of multilingual sentences of the model is still not as good as aligning solely the English sentences.

Notably, DistilRoBERTa-ALL of $MG_{SapSubset}$ with the setting of including SBERT-WK and using Google Translate corpus delivers the best precision and F-measure of all single `configs`. Without UMLS injection ($MG_{SBERTv2}$), RoBERTa-STSb using Google Translate and with SBERT-WK delivers the best precision, recall and F-measure. However, after UMLS injection, DistilRoBERTa-ALL is able to outperform RoBERTa-STSb in best precision and F-measure.

Table 11 Best single `configs` in recall within each Model-Group using only cosine similarity for ranking or additionally reranked with `Cross-Encoder`

Model	SBERT-WK	Corpus	Result size	Standard			`Cross-Encoder`		
				P	R	F	P	R	F
$MG_{SBERTv1}$									
$RoBERTa_{base}$-NS	With	GO	Top5	31.73	70.41	43.74	32.64	73.39	45.18
$RoBERTa_{large}$-NS	W/o	GO	Top5	31.86	70.32	43.85	31.67	71.22	43.84
$RoBERTa_{base}$-N	With	GO	Top5	31.35	69.59	43.23	31.79	71.49	44.01
$BERT_{base}$-N	With	GO	Top5	31.15	69.14	42.95	32.35	72.76	44.79
$RoBERTa_{base}$-NS	With	DL	Top5	30.91	68.51	42.60	32.39	72.85	44.85
$BERT_{base}$-N	With	OE	Top5	41.27	91.95	56.97	40.24	**90.50**	55.71
MG_{BPLM}									
$SapBERT\text{-}XLMR_{large}$	n.a.	GO	Top5	33.66	**73.67**	46.21	32.43	72.94	44.90
$SapBERT\text{-}XLMR_{large}$	n.a.	DL	Top5	33.17	72.49	45.51	32.56	73.21	45.07
$SapBERT\text{-}XLMR_{large}$	n.a.	MS	Top5	32.38	70.86	44.45	31.15	70.05	43.12
$SapBERT\text{-}XLMR_{base}$	n.a.	GO	Top5	31.91	69.59	43.76	32.27	72.58	44.68
$SapBERT\text{-}XLMR_{base}$	n.a.	MS	Top5	31.10	68.05	42.69	31.10	68.05	42.69
$SapBERT\text{-}XLMR_{large}$	n.a.	OE	Top5	41.89	91.58	57.48	39.96	89.86	55.32
$MG_{SBERTv2}$									
RoBERTa-STSb	With	GO	Top5	32.98	73.30	45.49	32.39	72.85	44.85
RoBERTa-STSb	With	DL	Top5	32.03	70.95	44.13	32.39	72.85	44.85
MPNet-STSb	With	GO	Top5	32.04	70.86	44.13	31.95	71.86	44.23
DistilRoBERTa-STSb	W/o	GO	Top5	31.70	70.32	43.70	32.56	73.21	45.07
MPNet-Paraphrase	W/o	GO	Top5	32.24	70.23	44.19	32.60	73.30	45.13
MPNet-STSb	With	OE	Top5	41.23	91.49	56.85	40.00	89.95	55.38
$MG_{SapFull}$									
RoBERTa-STSb	With	GO	Top5	33.09	73.57	45.65	33.16	74.57	45.91
MPNet-Paraphrase	W/o	GO	Top5	33.71	73.03	46.13	32.72	73.57	45.29
MPNet-STSb	W/o	GO	Top5	33.72	73.03	46.14	32.80	73.76	45.40
DistilRoBERTa-ALL	With	GO	Top5	33.02	72.85	45.44	32.52	73.12	45.01
MPNet-ALL	W/o	DL	Top5	33.40	72.22	45.68	32.23	72.49	44.62
RoBERTa-STSb	W/o	OE	Top5	42.18	**92.04**	57.85	40.08	90.14	55.49
$MG_{SapSubset}$									
RoBERTa-STSb	With	GO	Top5	33.05	73.48	45.59	33.28	**74.84**	46.07
DistilRoBERTa-ALL	With	GO	Top5	33.14	73.12	45.61	32.56	73.21	45.07
MPNet-ALL	W/o	GO	Top5	33.66	72.49	45.97	32.72	73.57	45.29
MPNet-ALL	With	GO	Top5	33.42	72.04	45.66	32.27	72.58	44.68
MPNet-ALL	W/o	DL	Top5	33.28	71.95	45.51	32.15	72.31	44.51
DistilRoBERTa-STSb	W/o	OE	Top5	41.71	91.76	57.35	40.04	90.05	55.43

In column SBERT-WK, "n.a." indicates that SBERT-WK is not applicable to BPLM models. The last row of each metric in grey is the best corresponding result obtained using the original English corpus (OE). The best recall of all `configs` of translated corpora and OE are in bold

GO Google Translate, *DL* DeepL, *MS* Microsoft Translator, *P* precision, *R* recall, *F* F-measure

RoBERTa-STSb delivers not only the best recalls among all SBERTv2 models, with the help of `Cross-Encoder`, it achieves the best recall of all models compared in this study. All the best metrics of the SBERT-based models (models in $MG_{SBERTv1}$, $MG_{SBERTv2}$, $MG_{SapFull}$ and $MG_{SapSubset}$) are produced by `configs` with SBERT-WK. This consents to our previous observation that adding SBERT-WK does improve annotation quality [1]. However, we also observe that many of the `configs` without SBERT-WK also perform well in recall and F-measure (Tables 11, 12). Furthermore, these best `configs` all achieve the best results using Google Translate corpora with only one exception (best precision of $MG_{SapFull}$). Consent to the AnnoMap results, the Google Translate is the most suitable machine translator also in the pretrained language model workflow.

The best precisions are delivered with `configs` of result size as Top1 and as expected, the best recalls of result size as Top5. The best F-measures are generated by `configs` with

Table 12 Best single `configs` in F-measure within each Model-Group using only cosine similarity for ranking or additionally reranked with `Cross-Encoder`

Model	SBERT-WK	Corpus	Result size	Standard			`Cross-Encoder`		
				P	R	F	P	R	F
$MG_{SBERTv1}$									
$RoBERTa_{base}$-N	With	GO	Top2	61.51	53.94	57.47	63.48	57.10	60.12
DistilBERT-NS	With	GO	Top2	61.09	53.85	57.24	63.38	57.01	60.03
$RoBERTa_{large}$-NS	W/o	GO	Top2	60.74	53.48	56.88	62.98	56.65	59.65
$RoBERTa_{base}$-NS	With	GO	Top2	60.78	53.30	56.80	63.58	57.19	60.22
$BERT_{base}$-N	With	GO	Top2	60.49	53.48	56.77	62.98	56.65	59.65
$RoBERTa_{base}$-NS	With	OE	Top2	86.36	75.66	80.66	81.59	73.39	77.27
MG_{BPLM}									
SapBERT-$XLMR_{large}$	n.a.	GO	Top2	62.75	54.57	58.37	63.38	57.01	60.03
SapBERT-$XLMR_{large}$	n.a.	DL	Top2	61.86	53.57	57.42	63.18	56.83	59.84
SapBERT-$XLMR_{large}$	n.a.	MS	Top2	59.92	51.95	55.65	59.96	53.94	56.79
SapBERT-$XLMR_{base}$	n.a.	GO	Top2	59.42	51.67	55.28	62.68	56.38	59.36
SapBERT-$XLMR_{large}$	n.a.	GO	Top3	48.02	62.62	54.36	47.48	64.07	54.55
SapBERT-$XLMR_{base}$	n.a.	OE	Top2	87.29	75.84	81.16	81.79	73.57	**77.47**
$MG_{SBERTv2}$									
RoBERTa-STSb	With	GO	Top2	62.54	55.29	58.69	63.78	57.38	60.41
MPNet-STSb	With	DL	Top2	61.65	54.12	57.64	63.58	57.19	60.22
DistilRoBERTa-STSb	W/o	GO	Top2	61.48	54.03	57.51	63.38	57.01	60.03
RoBERTa-STSb	With	DL	Top2	61.13	53.94	57.31	63.58	57.19	60.22
MiniLM(L12)-ALL	With	DL	Top2	61.21	53.85	57.29	63.48	57.10	60.12
MiniLM(L6)-ALL	W/o	OE	Top2	86.95	75.38	80.76	81.59	73.39	77.27
$MG_{SapFull}$									
DistilRoBERTa-ALL	With	GO	Top2	65.94	57.47	61.41	63.18	56.83	59.84
DistilRoBERTa-ALL	With	MS	Top2	65.63	57.19	61.12	61.07	54.93	57.84
RoBERTa-STSb	With	GO	Top2	64.33	56.47	60.14	64.19	57.74	60.79
DistilRoBERTa-ALL	With	DL	Top2	63.73	55.66	59.42	63.48	57.10	60.12
MPNet-STSb	W/o	GO	Top2	63.73	55.02	59.06	63.38	57.01	60.03
MiniLM(L6)-Paraphrase	W/o	OE	Top2	87.83	75.75	**81.34**	81.49	73.30	77.18
$MG_{SapSubset}$									
DistilRoBERTa-ALL	With	GO	Top2	66.46	57.92	**61.90**	63.78	57.38	60.41
DistilRoBERTa-ALL	With	MS	Top2	64.80	56.47	60.35	60.36	54.30	57.17
RoBERTa-STSb	With	GO	Top2	64.43	56.56	60.24	64.39	57.92	**60.98**
DistilRoBERTa-ALL	With	DL	Top2	63.21	55.20	58.94	63.18	56.83	59.84
MPNet-ALL	W/o	DL	Top2	63.56	54.93	58.93	63.18	56.83	59.84
MiniLM(L6)-Paraphrase	W/o	OE	Top2	87.55	75.75	81.22	81.29	73.12	76.99

In column SBERT-WK, "n.a." indicates that SBERT-WK is not applicable to BPLM models. The last row of each metric in grey is the best corresponding result obtained using the original English corpus (OE). The best F-measure of all `configs` of translated corpora and OE are in bold

GO Google Translate, *DL* DeepL, *MS* Microsoft Translator, *P* precision, *R* recall, *F* F-measure

Top2 as result size. Only 2 of the 75 `configs` in Tables 10 - 12 are exceptions. They are the `config` SapBERT-$XLMR_{large}$ using Google Translate ranked 5th in both the precision table (result size = 2 instead of 1) and the F-measure table (result size = 3 instead of 2). These exceptions are mainly due to that the SapBERT-$XLMR_{large}$ performs the best in the MG_{BPLM} and therefore, even with sub-optimal result size, it still outperforms other `configs`. The `configs` with Top2 result size generates the best F-measure can be explained by that in our corpus most of the questions have 2 GSC annotations (as shown in Fig. 2).

Computation efficiency We used two NVIDIA V100 Tensor Cores as GPUs to encode the questions and the UMLS concepts. Table 13 presents the computation time of the models in $MG_{SBERTv1}$, MG_{BPLM} and $MG_{SBERTv2}$. Since the UMLS injection using SAP does not change the model

Table 13 Computing time per embedding of models in each Model-Group

Model	W/o SBERT-WK (ms)	With SBERT-WK (ms)
$MG_{SBERTv1}$		
$BERT_{base}$	0.99	46.47
$BERT_{large}$	1.48	n.a.
$RoBERTa_{base}$	0.79	35.92
$RoBERTa_{large}$	1.37	n.a.
DistilBERT	0.55	12.78
MG_{BPLM}		
BioBERT	19.54	n.a.
PubMedBERT	19.43	n.a.
SapBERT	19.64	n.a.
CODER	19.79	n.a.
$MG_{SBERTv2}$		
MPNet	0.51	34.17
RoBERTa	0.51	33.24
DistilRoBERTa	0.30	22.59
MiniLM(L12)	0.46	33.78
MiniLM(L6)	0.27	22.55

"n.a." indicates that SBERT-WK is not applied to the large models and the BPLM models

structure, the encoding time of the models in $MG_{SapFull}$ and $MG_{SapSubset}$ remains the same as those in $MG_{SBERTv2}$. Similarly, the same model but pretrained on different corpora (e.g., MPNet-STSb and MPNet-ALL) also have the same computation time and therefore are not shown separately in the table. We can conclude that the newly selected SBERTv2 models not only outperform the SBERTv1 models in annotation quality, they are also more efficient. MiniLM(L6) and DistilRoBERTa are the fastest models. Applying SBERT-WK drastically increases the computation time because it relies on CPU to operate. Owing to the fact that all BPLM models are direct derivatives of the initial BERT, their efficiency are alike. They are approximately 10% faster than the fastest SBERTv2 models with SBERT-WK.

Combination of results In our previous study [1] we showed that combining using set operations on the result sets of different translated corpora can improve annotation quality further. Therefore, we applied the combinations and obtained the best precisions by intersecting the result sets (Table 14), the best recalls by union (Table 15) and the best F-measures by 2-vote-agreements (Table 16). In each table, we present the best combination result of each Model-Group on the given metric. The last three columns of these tables also show the results of reranking the candidates using `Cross-Encoder` before combination.

Table 14 The best combination results in precision by combining three different corpora within each Model-Group using intersection

Model	SBERT-WK	Corpus	Result size	Standard			`Cross-Encoder`		
				P	R	F	P	R	F
$MG_{SBERTv1}$									
DistilBERT-N	With	MS	Top1 (Top2)	**93.46**	9.05	16.50	82.80	44.43	57.83
$RoBERTa_{base}$-NS	With	GO							
$RoBERTa_{large}$-N	W/o	DL							
MG_{BPLM}									
$CODER_{ALL}$	n.a.	MS	Top1 (Top2)	92.94	7.15	13.28	**87.32**	38.64	53.58
SapBERT-$XLMR_{large}$	n.a.	DL							
$SapBERT_{mean}$	n.a.	GO							
$MG_{SBERTv2}$									
MPNet-STSb	With	GO	Top1 (Top2)	93.00	8.42	15.44	86.30	39.91	54.58
DistilRoBERTa-STSb	With	MS							
MiniLM(L12)-Paraphrase	W/o	DL							
$MG_{SapFull}$									
MPNet-STSb	W/o	DL	Top1 (Top2)	90.62	10.50	18.82	85.35	30.59	45.04
RoBERTa-STSb	With	GO							
MPNet-STSb	With	MS							
$MG_{SapSubset}$									
RoBERTa-STSb	With	GO	Top1 (Top2)	91.26	8.51	15.56	84.18	29.86	44.09
MiniLM(L6)-Paraphrase	With	DL							
DistilRoBERTa-STSb	With	MS							

In column SBERT-WK, "n.a." indicates that SBERT-WK is not applicable to BPLM models. The result size of `Cross-Encoder` results are in brackets

GO Google Translate, *DL* DeepL, *MS* Microsoft Translator, *P* precision, *R* recall, *F* F-measure

Table 15 The best combination results in recall by combining three different corpora within each Model-Group using union

Model	SBERT-WK	Corpus	Result size	Standard			Cross-Encoder		
				P	R	F	P	R	F
$MG_{SBERTv1}$									
$RoBERTa_{base}$-NS	With	DL	Top5 (Top5)	19.94	84.16	32.24	26.01	**84.16**	39.74
$RoBERTa_{large}$-NS	W/o	MS							
DistilBERT-NS	With	GO							
MG_{BPLM}									
$SapBERT\text{-}XLMR_{base}$	n.a.	MS	Top5 (Top5)	20.43	83.98	32.86	23.75	82.26	36.86
$SapBERT\text{-}XLMR_{large}$	n.a.	DL							
$SapBERT_{CLS}$	n.a.	GO							
$MG_{SBERTv2}$									
MiniLM(L6)-ALL	W/o	DL	Top5 (Top5)	20.21	84.89	32.65	23.86	81.99	36.96
RoBERTa-STSb	With	GO							
MPNet-STSb	W/o	MS							
$MG_{SapFull}$									
MPNet-Paraphrase	W/o	GO	Top5 (Top5)	22.59	85.16	35.71	23.66	82.81	36.80
RoBERTa-STSb	With	DL							
MPNet-ALL	W/o	MS							
$MG_{SapSubset}$									
RoBERTa-STSb	With	GO	Top5 (Top5)	22.43	**85.25**	35.51	23.20	81.63	36.13
MPNet-ALL	W/o	DL							
MPNet-STSb	W/o	MS							

In column SBERT-WK, "n.a." indicates that SBERT-WK is not applicable to BPLM models. The result size of Cross-Encoder results are in brackets

GO Google Translate, *DL* DeepL, *MS* Microsoft Translator, *P* precision, *R* recall, *F* F-measure

Overall, we are able to improve the best precision of using translated corpora to 93.46% by combining the results of the $MG_{SBERTv1}$ models (Table 14). This is an improvement of 22.23% compared to best single config result (71.23%, Table 10). Combining three models in $MG_{SapSubset}$ achieves the best recall of 85.25% (Table 15), an increase of 11.58% compared to the best single config (73.67%, Table 11). On the other hand, combination only raises the best F-measure of 1.84% compared to single config (from 61.90 to 63.74%). Again, this best F-measure result is delivered by combining models in $MG_{SapSubset}$ as in best recall result. Unlike the enhancement we could see in the single config results, reranking using Cross-Encoder can not improve the combination results further but rather worsens them.

Recall vs result size It is clear, that applying union to three config result sets achieves a higher recall than a single config, as after union the result size increases by a factor of three at the most. Hence, we ask, given the same result size, which can deliver better recall: the union of three configs or a single config? To answer this question, we plot the change of recall over increasing result size up to 150. We only consider the best model regarding the metric recall of each Model-Group (Fig. 4). We observe that the increase of recalls flattens at a result size of approximately 13 when annotating the original English corpus (Fig. 4a). On the other hand, when annotating a translated corpus, the recalls keep increasing even until a result size of 140, though the increasing rates mostly reaches a saturation at the result size between 55 and 75 (Fig. 4b). Moreover, the single configs deliver higher recalls than combination when the result sizes smaller than approximately 30. However, with larger result sizes, the recalls of combination overtake those of single configs. This shows combination does raise the overall recall limit compared to single configs. These plots also reveal the potential maximum recalls can be reached when retaining the best 150 candidates. By combining the $MG_{SapSubset}$ models, it is possible to reach a recall of 94.48% and with single config using the best BPLM model (i.e., $SapBERT\text{-}XLMR_{large}$), a recall of 93.48% is attainable.

Result Summary

We compile the best results generated by each approach and show them in Fig. 5. Notably, if the task is not cross-lingual but to annotate the original English forms, the proposed *Workflow-MT* still outperforms the traditional string matching method even the questions and the corresponding concepts are syntactically identical. Using the sentence

Table 16 The best combination results in F-measure by combining three different corpora within each Model-Group using 2-vote-agreement

Model	SBERT-WK	Corpus	Result size	Standard			`Cross-Encoder`		
				P	R	F	P	R	F
$MG_{SBERTv1}$									
$BERT_{base}$-N	With	MS	Top2 (Top2)	77.35	52.85	62.80	70.49	53.39	60.76
$RoBERTa_{large}$-N	W/o	DL							
DistilBERT-NS	With	GO							
MG_{BPLM}									
SapBERT-$XLMR_{large}$	n.a.	GO	Top2 (Top2)	68.78	53.03	59.89	69.36	53.67	60.51
SapBERT-$XLMR_{large}$	n.a.	DL							
SapBERT-$XLMR_{base}$	n.a.	MS							
$MG_{SBERTv2}$									
MPNet-Paraphrase	With	MS	Top2 (Top3)	74.97	54.75	63.28	68.20	56.47	61.78
MPNet-STSb	With	DL							
RoBERTa-STSb	With	GO							
$MG_{SapFull}$									
DistilRoBERTa-ALL	With	MS	Top2 (Top5)	74.97	54.75	63.28	58.67	57.56	58.11
MiniLM(L12)-ALL	With	GO							
MPNet-ALL	W/o	DL							
$MG_{SapSubset}$									
MiniLM(L12)-ALL	W/o	MS	Top2 (Top3)	74.40	55.75	**63.74**	68.32	57.38	**62.37**
RoBERTa-STSb	With	DL							
DistilRoBERTa-ALL	With	GO							

In column SBERT-WK, "n.a." indicates that SBERT-WK is not applicable to BPLM models. The result size of `Cross-Encoder` results are in brackets

GO Google Translate, *DL* DeepL, *MS* Microsoft Translator, *P* precision, *R* recall, *F* F-measure

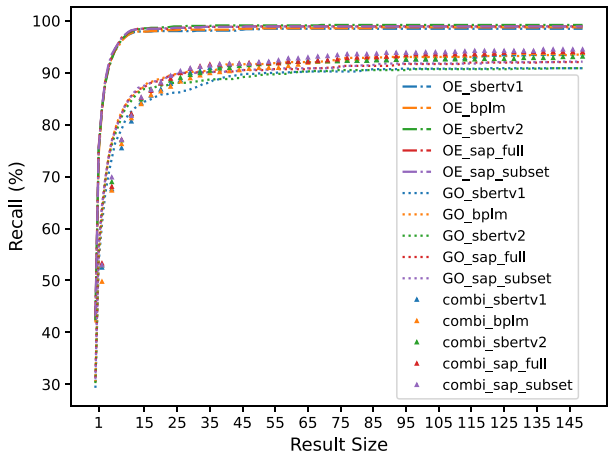
encoder workflows, we gain a large improvement in recall and F-measure. This indicates that the use of sentence embeddings as semantic representation does help to find many more correct concepts. Incorporating machine translators into the workflow (*Workflow-MT*) produces better results than using original German forms as input (*Workflow-Multi*). This observation still holds true when the same encoder is applied in different workflows, as we have seen on SapBERT-$XLMR_{large}$. Hence, we can conclude that using machine translator is still inevitable before better aligned multilingual sentence encoders are available.

The best annotation quality we achieve using single `config` is 71.23% in precision, 74.84% in recall and 61.90% in F-measure. All these best results are generated using UMLS subset injected SBERTv2 models, i.e., DistilRoBERTa-ALL for precision and F-measure and RoBERTa-STSb for recall. Further, these two models are both enhanced by SBERT-WK and used Google Translate corpus as input. In addition, RoBERTa-STSb obtains the best recall with reranking of `Cross-Encoder`.

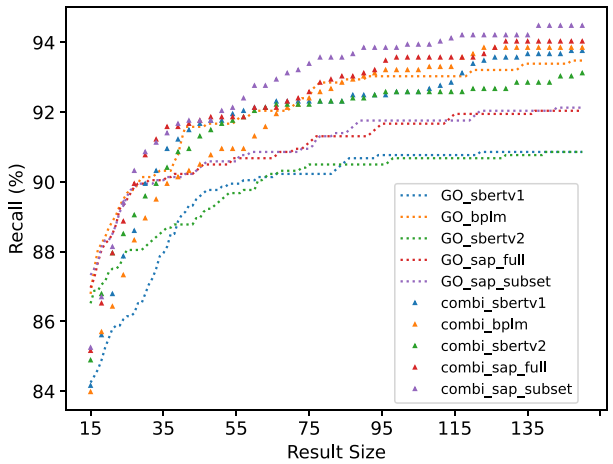
Figure 5 also shows that among the three metrics, precision benefits most from combination. The best precision, 93.46%, on translated corpora using combination, is almost equivalent to the result of using AnnoMap to annotate the original English corpus (93.6%). Similarly, we also achieve a best recall of 85.25% that is also amounting to that of the AnnoMap on original English corpus (86%). However, there is still space for improvement in terms of F-measure (the best F-measure 63.74%). Overall, we achieved an improvement of 136% in recall (from 36.11% of AnnoMap to 85.25% of combination), 52% in precision (61.60% of AnnoMap to 93.46% of combination) and 66% in F-measure (AnnoMap: 38.36%, combination: 63.74%). We set our maximum result size as Top5 so that the system can provide a reasonably short list of candidates for further manual verification (semi-automatic annotation). In this case, with the best recall of 85.25% and a precision of 100%, a F-measure of 92.04% is plausible.

Conclusion

In this study, we apply BERT-based pretrained language models to generate sentence embeddings to solve cross-lingual biomedical concept normalization problem. We show that the annotation quality can be improved significantly compared to the conventional string matching tool. For the future work, we aim to apply such techniques onto other types of annotations (e.g., biomedical name entities) or in other domains.



(a)



(b)

Fig. 4 Increase of recall against result size. **a** The result of the best single `config` or combination in recall of each Model-Group starts from result size of 1, including the results of annotating original English (OE) corpus. **b** The same results without OE corpus and starts from result size of 15

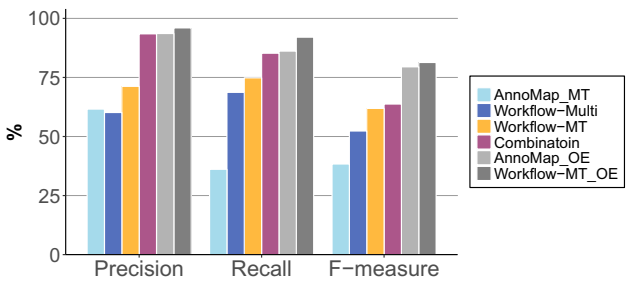


Fig. 5 The best results of each approach. *OE* original English corpus

We select current SOTA models that are specifically pretrained on biomedical corpus (the BPLM models) or only pretrained on plain English text (the SBERT models without UMLS injection). The results show that the best performance of these two types of models is similar. This can be due to that many of the questions in our medical forms are in colloquial language as they are designed to interview general public. Therefore, extra biomedical corpus pretraining does not benefit the annotation results. Furthermore, we show that we can further enhance the annotation quality of SBERTv2 models using UMLS injection and outperform the best BPLM model (which is already UMLS-injected). We also discover that UMLS injection using only the relevant subset is sufficient to produce comparable (or even slightly better) results than using the full version of the UMLS. This observation is similar to the idea of the PubMedBERT [24] that more pretraining using out-domain corpora is not necessarily beneficial for solving domain-specific tasks.

We tested two post-processing strategies in this study. Combination can improve annotation quality significantly and also raise the recall upper bound compared to single `config`. The reranking of `Cross-Encoder` benefits the results of `configs` but does not improve combination result further. However, as Fig. 4 shows, with the result size of 150, we have the potential of finding up to 94.48% of the correct annotations. Hence, we plan to develop better post-processing approaches that can rerank the candidates so that the correct annotations are included in the Top5 result sets.

Acknowledgements This work is funded by the German Research Foundation (DFG) (grant RA 497/22-1, "ELISA - Evolution of Semantic Annotations") and by the ScaDS.AI. We also thank Leipzig University Computing Centre for providing the computation resources and support.

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

References

1. Lin Y-C, Hoffmann P, Rahm E. Enhancing cross-lingual semantic annotations using deep network sentence embeddings. In: Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021), vol. 5. HEALTHINF; 2021. p. 188–99.
2. Spitzer RL, Kroenke K, Williams JB, Löwe B. A brief measure for assessing Generalized Anxiety Disorder: the GAD-7. Arch Intern Med. 2006;166(10):1092–7.
3. Sousa TV, Viveiros V, Chai MV, Vicente FL, Jesus G, Carnot MJ, Gordo AC, Ferreira PL. Reliability and validity of the Portuguese version of the Generalized Anxiety Disorder (GAD-7) scale. Health Qual Life Outcomes. 2015;13(1):50.
4. Löwe B, Decker O, Müller S, Brähler E, Schellberg D, Herzog W, Herzberg PY. Validation and standardization of the Generalized Anxiety Disorder Screener (GAD-7) in the general population. Med Care. 2008;46(3):266–74.
5. García-Campayo J, Zamorano E, Ruiz MA, Pardo A, Pérez-Páramo M, López-Gómez V, Freire O, Rejas J. Cultural adaptation into Spanish of the Generalized Anxiety Disorder-7 (GAD-7) scale as a screening tool. Health Qual Life Outcomes. 2010;8(1):8.
6. Lin Y-C, Christen V, Groß A, Kirsten T, Cardoso SD, Pruski C, Da Silveira M, Rahm E. Evaluating cross-lingual semantic annotation for medical forms. In: HEALTHINF; 2020. p. 145–55.
7. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. 2018. arXiv preprint arXiv:1810.04805.
8. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Advances in neural information processing systems. 2017. pp. 5998–6008.
9. Zhu Y, Kiros R, Zemel R, Salakhutdinov R, Urtasun R, Torralba A, Fidler S. Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE International Conference on Computer Vision, 2015. pp. 19–27.
10. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. RoBERTa: a robustly optimized BERT pretraining approach. 2019. arXiv preprint arXiv:1907.11692.
11. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2019. arXiv preprint arXiv:1910.01108.
12. Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006. pp. 535–41.
13. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. 2015. arXiv preprint arXiv:1503.02531.
14. Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M. MiniLM: Deep self-attention distillation for task-agnostic compression of pretrained transformers. In: NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020; p.5776–5788
15. Mirzadeh SI, Farajtabar M, Li A, Levine N, Matsukawa A, Ghasemzadeh H. Improved knowledge distillation via teacher assistant. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020;34:5191–98.
16. Sung M, Jeon H, Lee J, Kang J. Biomedical entity representations with synonym marginalization. 2020. arXiv preprint arXiv:2005.00239.
17. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV. XLNet: generalized autoregressive pretraining for language understanding. In: NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019. pp. 5753–5763
18. Reimers N, Gurevych I. Sentence-BERT: sentence embeddings using siamese BERT-networks. 2019. arXiv preprint arXiv:1908.10084.
19. Wang B, Kuo C-CJ. SBERT-WK: a sentence embedding method by dissecting BERT-based word models. 2020. arXiv preprint arXiv:2002.06652.
20. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics. 2020;36(4):1234–40.
21. Alsentzer E, Murphy JR, Boag W, Weng W-H, Jin D, Naumann T, McDermott M. Publicly available clinical BERT embeddings. 2019. arXiv preprint arXiv:1904.03323.
22. Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, Mark RG. MIMIC-III, a freely accessible critical care database. Sci Data. 2016;3(1):1–9.
23. Peng Y, Yan S, Lu Z. Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMO on ten benchmarking datasets. 2019. arXiv preprint arXiv:1906.05474.
24. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J, Poon H. Domain-specific language model pretraining for biomedical natural language processing. 2020. arXiv preprint arXiv:2007.15779.
25. Hao B, Zhu H, Paschalidis I. Enhancing clinical BERT embedding using a biomedical knowledge base. In: Proceedings of the 28th International Conference on Computational Linguistics, 2020. pp. 657–61.
26. Liu F, Shareghi E, Meng Z, Basaldella M, Collier N. 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, p. 4228–38. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.naacl-main.334. https://aclanthology.org/2021.naacl-main.334.
27. Yuan Z, Zhao Z, Yu S. CODER: knowledge infused cross-lingual medical term embedding for term normalization. 2020. arXiv preprint arXiv:2011.02947.
28. Yuan Z, Liu Y, Tan C, Huang S, Huang F. Improving biomedical pretrained language models with knowledge. 2021. arXiv preprint arXiv:2104.10344.
29. Wang X, Han X, Huang W, Dong D, Scott MR. Multi-similarity loss with general pair weighting for deep metric learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. pp. 5022–30.
30. Reimers N, Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation. 2020. arXiv preprint arXiv:2004.09813.
31. Conneau A, Lample G, Ranzato M, Denoyer L, Jégou H. Word translation without parallel data. 2018. arXiv preprint arXiv:1710.04087.
32. Loeffler M, Engel C, Ahnert P, Alfermann D, Arelin K, Baber R, Beutner F, Binder H, Brähler E, Burkhardt R, et al. The LIFE-Adult-Study: objectives and design of a population-based cohort study with 10,000 deeply phenotyped adults in Germany. BMC Public Health. 2015;15(1):691.
33. Kroenke K, Spitzer RL, Williams JBW. The PHQ-15: validity of a new measure for evaluating the severity of somatic symptoms. Psychosom Med. 2002;64(2):258–66.
34. Christen V, Groß A, Varghese J, Dugas M, Rahm E. Annotating medical forms using UMLS. In: International Conference on Data Integration in the Life Sciences. Springer; 2015. p. 55–69.
35. Christen V, Groß A, Rahm E. A reuse-based annotation approach for medical documents. In: International Semantic Web Conference. Springer; 2016. p. 135–50.
36. Bowman SR, Angeli G, Potts C, Manning CD. A large annotated corpus for learning natural language inference. 2015. arXiv preprint arXiv:1508.05326.

37. Williams A, Nangia N, Bowman SR. A broad-coverage challenge corpus for sentence understanding through inference. 2017. arXiv preprint arXiv:1704.05426.
38. Cer D, Diab M, Agirre E, Lopez-Gazpio I, Specia L. Semeval-2017 task 1: semantic textual similarity-multilingual and cross-lingual focused evaluation. 2017. arXiv preprint arXiv:1708.00055.
39. Lin Y-C, Christen V, Groß A, Cardoso SD, Pruski C, Da Silveira M, Rahm E. Evaluating and improving annotation tools for medical forms. In: Proc. Data Integration in the Life Science (DILS 2017). Springer; 2017. pp. 1–16.
40. Humeau S, Shuster K, Lachaux M-A, Weston J. Poly-encoders: transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. 2019. arXiv preprint arXiv:1905.01969.
41. Thakur N, Reimers N, Daxenberger J, Gurevych I. Augmented SBERT: data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. 2020. arXiv preprint arXiv:2010.08240.