Learning Knowledge Representation by Aligning Text and Triples via Finetuned Pretrained Language Models

Víctor Jesús Sotelo Chico[a] and Julio Cesar dos Reis[b]
Instituto de Computação, Universidade Estadual de Campinas (UNICAMP), Brazil

Keywords: Language Model, Knowledge Graph, Multimodal Encoders.

Abstract: Representation learning has produced embedding for structure and unstructured knowledge constructed independently, not sharing a vectorial space. Alignment between text and RDF triples has been explored in natural language generation, from RDF verbalizers to generative models. Existing approaches have treated the semantics in these data via unsupervised approaches proposed to allow semantic alignment with adequate application studies. The existing datasets involved in text-triples are limited and have only been applied to text-to-triple generation rather than for representation. This research proposes a supervised approach for representing triples. Our approach feeds an existing pretrained model with triple-text pairs exploring measures for the semantic alignment between the pair elements. Our solution employs a data augmentation technique with contrastive loss to address the dataset limitation. We applied a loss function that requires only positive examples, which is suitable for the explored dataset. Our experimental evaluation measures the effectiveness of the fine-tuned models in two main tasks: 'Semantic Similarity' and 'Information Retrieval'. These tasks were addressed to measure whether our designed models can learn triple representation while maintaining the semantics learned by the text encoder models. Our contribution paves the way for better embeddings targeting text-triples alignment without huge data, bridging unstructured text and knowledge graph data.

1 INTRODUCTION

The evolution of natural language processing in recent years has been driven by the introduction of large language models (LLMs) (Min et al., 2023), demonstrating the ability to process a large amount of text. They seem to understand human language properly, and their main application is suited to developing more powerful question-and-answer systems. In these applications, LLMs suffer problems with hallucinations (Perković et al., 2024) when they answer with no accurate information. This behavior can be explained by the nature of knowledge acquired by LLMs from a large corpus of unstructured text, which might not be appropriately curated.

Existing research has investigated the possibility of integrating structure knowledge with LLMs. For instance, Pan *et al.* (Pan et al., 2024) proposed a roadmap to incorporate Knowledge Graphs (KG) with LLMs to overcome hallucination issues, taking advantage of the structure and modeling of KGs and their interpretability.

Working with LLM-KG means dealing with their input elements (text and triples), which are heterogeneous data underlying different semantics. One of the most well-known tasks that work with aligning these data is the task of Natural Language Generation (NLG) in converting triples into natural text (Lapalme, 2020; Zhu et al., 2019) and natural text into triples (Regino et al., 2023). These applications cover the aligning triple to text to create one of them. Existing projects make datasets available for this challenge (Castro Ferreira et al., 2020).

We understand that existing approaches related to NLG need to discuss the semantics of text and triple in terms of embedding representations. Constructing embeddings representing the text develops further and achieves new state-of-the-art with the introduction of Transformers pre-trained model to represent text (Xia et al., 2020), in which a text is mapped to a numerical vector. Similarly, Knowledge graph embeddings (KGE) (Cao et al., 2024) create a vector representation for their elements, entities, and relations.

Representing triples as a single embedding has been investigated in the literature (Fionda and Pirrò,

[a] https://orcid.org/0000-0001-9245-8753
[b] https://orcid.org/0000-0002-9545-2098

Chico, V. and Reis, J.
Learning Knowledge Representation by Aligning Text and Triples via Finetuned Pretrained Language Models.
DOI: 10.5220/0013015100003838

In *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2024) - Volume 2: KEOD*, pages 51-62
ISBN: 978-989-758-716-0; ISSN: 2184-3228

2020; Kalinowski and An, 2022) due to their importance in triples classification, clustering, and recommendation systems. Generating vectors to represent triplets (as a whole) instead of separate vectors for specific triple elements (entities and relations) alters the representation of elements in a KG in a vector space. We understand that such an approach is further adequate and advances representation learning.

These developments have achieved new, recent, state-of-the-art encoders (Patil et al., 2023). However, investigating encoders capable of representing facts, such as triples and texts, has followed independent paths. This entails separating two approaches that define the knowledge. Without a common representation for such knowledge, intermediate spaces, such as using SPARQL queries (generated by humans or generative models), are necessary. Treating languages such as RDF as texts with a proper knowledge representation in vector spaces might result in poor results using incorrect information.

This article presents an investigation into developing a prevalent method for convergent semantic alignment using a small dataset with triples and natural texts. Our study develops a more robust solution capable of representing knowledge in the same vector space, considering KG triples and instructed texts in an integrated way. Our approach covers the task of using pre-trained language models and finetuning them using parallel data of triples and their respective equivalents in natural texts. To improve the alignment quality, we proposed a data augmentation technique over the small dataset WEBNLG (Gardent et al., 2017), which provides examples of triple with their respective meaning in natural text. The developed solution aims to guarantee the preservation of semantic representation.

We evaluate our proposal by considering different NLP tasks, such as Semantic Textual Similarity and Information Retrieval (IR). The first collects and assesses textual data, processing their embedding over the STS-22 dataset and detecting if the finetuning process degrades the STS benchmark. Finally, IR creates embedding for triples and text from WEBNLG. We conduct two main IR configurations: (1) we recover text using a triple as a query, and then (2) we use text to recover the adequate triples.

Our results demonstrate the need for improved alignment between text and triples instead of applying these pretrained models directly because such an approach only considers grammar sharing and not a genuine semantic alignment. We found that finetuning improved model effectiveness, particularly in the retrieval task (MRR@1 score). Furthermore, using Contrastive loss and augmentation enabled effective semantic learning with models without losing effectiveness in the semantic similarity task. Our findings establish that learning the alignment between text and triples is possible only using positive examples (pairs of examples with the same meaning) in small datasets.

The remainder of this article is organized as follows: Section 2 presents the background concepts and discusses related studies associated with our investigation; Section 3 presents our proposed original method to construct a semantic model alignment for text and triples. Section 4 presents our experimental evaluation methodology based on two evaluation scenarios. Section 5 reports on our obtained results, whereas Section 6 discusses our findings and study strengths; finally, Section 7 wraps up the article and points out directions for future investigations.

2 LITERATURE REVIEW

Subsection 2.1 addresses background concepts relevant to our study, and Subsection 2.2 presents a synthesis of related studies to our proposal.

2.1 Background Concepts and Techniques

Language Models. A Language model (Chang and Bergen, 2024) is a probabilistic model that learns language properties from unstructured text trained in a task that does not require human annotation. These trained tasks provide the underlying model knowledge about how human languages are represented by understanding how text is composed. This learns from raw data rather than relying on annotating all the grammar and syntax properties.

Large Language Models (LLM). LLM (Min et al., 2023) stands for massive artificial intelligence models with billions of parameters that can understand human language and perform NLP tasks such as writing, summarizing, and others. These models improve traditional Language models, such as *BERT* (Devlin et al., 2019), which require a finetuned process over an specific task.

Text Embeddings. Creating a vector representation for the meaning of the texts has been developed since the beginning of Natural Language Processing (Patil et al., 2023). This includes techniques such as Word2vec (Mikolov et al., 2013), which map words into a unique vector, to pre-trained language models such as BERT (Devlin et al., 2019) that considers the surrounding context around the words to assign a vector and applies subword tokenization to avoid mapping each word to a numerical id. The development of language models allowed the construction of more robust embedding models, which could create different vector representations for a word depending on the surrounding words; we refer to this as context because it determines the vector creations. This represents an advancement over the initial approaches, such as a bag of words (Qader et al., 2019) by solving a problem of polysemy (having the exact words with different meanings).

Knowledge Graphs. KGs (Ji et al., 2022) are directed graphs that model real-world facts. Each node represents an entity in such graphs, and the edges represent a relation between them. Formally, we define a KG as a set of Entities (E), relations (R), and triples (T), such as:

$$KG = \{E, R, T\}, e_i \in E, r_i \in R, t_j \in T$$

$$t_j = (e_x, r_y, e_z)$$

For example, the fact *"The Star Wars film was directed by George Lucas"* can be represented by the triple $(starWarsFilm, directed, GeorgeLucas)$. Over the same KG, we can obtain more triples with different facts about other relevant information, such as the film's release date. KG requires knowledge modeling, giving a well-defined and correct semantic definition for concepts in a given domain. This is accomplished by using an ***Ontology*** as an artifact computationally representing knowledge (Ding et al., 2007). KG provides an explanation and reasoning over the inferences extracted from the graphs.

Knowledge Graph Embeddings (KGE). The structured knowledge encoded in KGs requires a query language such as SPARQL (Hogan, 2020) to process and extract information from it. This language allows the creation of specific queries to extract information from KGs. However, this creates a new issue that requires either manually creating these queries or using generative models to create them..To overcome this problem, the KG embedding (KGE) (Yan et al., 2022) aims to represent KG components (entities and relations) in a semantic vector form. Such vectors aim to describe the semantics involved in knowledge modeling accurately. In this sense, similar entities must be near distances between their embeddings. The main application of KG embeddings is creating an alternative format for computer-consuming KG in machine learning models. This makes it possible to apply KG to link prediction tasks, triplet classification, recommendation systems, and others (Wang et al., 2017).

2.2 Related Studies

Lapalme *et al.* (Lapalme, 2020) proposed an English RDF verbalizer that uses a symbolic approach to process an RDF-Triple, extracting the subject and predicate corresponding to the subject and object in a sentence. The predicate is then mapped to a verb phrase, which determines the structure of the final sentence. Human evaluation showed that this approach needs more fluency and that the human work involved in the application is difficult to scale for more customized applications.

Similarly, Abhishek *et al.* (Abhishek et al., 2022) proposed a technique to address triples to text generation with the addition that they cover a scenario different from English languages. This demands more datasets for non-English languages, so they created XAlign, a multilingual dataset. These were constructed from facts to text generation, aligning triple facts with natural text. Their work demonstrated the relevance of creating triples and text alignment for low-resource languages to reduce human efforts.

The other direction can also be explored (from text to triples instead of triple to text). The study conducted by Regino *et al.* (Regino et al., 2023) developed a framework called QART for generating RDF-Triples from E-commerce Product Question Answering using an E-commerce dataset and pretrained LM and LLM application in few-shot learning.

Daw *et al.* (Daw et al., 2021) proposed the alignment of English triples to Hindi sentences using NER-based filtering that uses semantic similarity. Their approach maps Hindi and English into the same vector spaces to recognize the most relevant words in English for a given word in Hindi. They used a key phrase extraction to extract the critical phrase from Hindi text and then applied POS-tag-based heuristics. Then, the similarity is based on the key phrase and the triples. This approach creates a heuristic for allowing the alignment between text and triples. However, these intermediate steps map all words into the same languages without providing training in creating semantic alignment to the embedding spaces.

Moreover, Pahuja *et al.* (Pahuja et al., 2021) proposed a method for aligning knowledge bases (KB) with texts; the authors used data from Wikidata for the structure knowledge and Wikipedia for unstructured. This enables the possibility of testing the proposal's alignment methods. They considered alignment using the same embedding. This involves mapping the entities from KB to a vector space for text data.

A synergy between LM and KG has been explored; Zhu *et al.* (Zhu et al., 2023) focused on finding a unique representation creating a heterogeneous language model trained based on unstructured, semi-structured, and structured data. The work was developed by training with a text corpus of tourism websites and constructing a KG for a tourism context using websites. They proposed a pre-trained model that handles the formats of text, as well as triples. The authors proposed a different training objective: A mock language model, title matching (whether a title matches a paragraph), and triple classification for unstructured, semistructured, and structured, respectively. As a consequence, these data were mapped into the exact contextual representation. A primary aspect of their study is that they do not use KGE directly. They use it to prepare their training objectives in pretrained models. For example, they train a model to perform the triple classification task.

Table 1 summarizes our related work review. We identified existing studies that focus on alignment for NLG tasks and studies that handle semantic representation require training from scratch, which is computationally expensive.

Our present study focuses on the challenge of dealing with diverse data using various methods. Compared to other researchers, we are not addressing the issue by using a pretrained encoder directly, employing intermediate steps to map triples, or creating pretrained models from scratch, which is often unsuitable. To the best of our knowledge, the study of the application of text to triple alignment needs to investigate whether the application of pre-trained encoders is suitable for unsupervised alignments. Moreover, a pre-trained model's tuning process can change the pretrained models' semantic learning (capability to understand the text and adequately recover a similar one).

Our study complements the presented investigations by asking how well and acceptable the current strategies of the used pre-trained models to align triples and text are for direct application to vector representation work. Additionally, we attempt to determine whether it is possible to create embedding representations using small-size datasets. We propose supervised training in small datasets for fine-tuning currently open state-of-the-art models for semantic representation between text and triples. We construct a triple vector representation while maintaining adequate text vector representation for the downstream NLP applications. This aims to map vectors and triples into the same vector spaces obtained by pretrained models. This might enhance current triple alignment literature and open the field for future research, which can help create robust triple-text alignment.

3 SUPERVISED SEMANTIC ALIGNMENT FOR TRIPLE-TEXT EMBEDDING CONSTRUCTION

We present our proposal for aligning text and triples using supervised learning. Subsection 3.1 presents the datasets explored in our study. Subsection 3.2 describes the models used in the finetuning process. Subsection 3.4 presents the designed specific procedure to conduct the learning representation process and its outcome by involving our specific decisions.

3.1 Datasets

WebNLG. We chose *WebNLG* (Gardent et al., 2017), a dataset from a challenge competition to transform triples into text, and to the best of our knowledge, it is the only one that aligned triples and text. This dataset comprises examples of triples and their equivalent in natural language text. The collection of triples can be expressed in the natural text; for example, the given two triples (Leonardo_da_Vinci, Profession, Painter) and (Leonardo_da_Vinci, Born, Italy) can be expressed in natural text with the sentences 'Leonardo da Vinci is an Italian painter.'

In this study, we focus on establishing a one-to-many relationship between a single triple and multiple texts to ensure each triple is aligned with a unique text.

STS-12. We selected the *STS-12* dataset for the semantic textual similarity (STS) task, composed of sentence pairs labeled with values from 0 to 5, in which five expresses higher similarity. This dataset was selected to maintain the model's capability to handle only textual data.

Table 2 presents the distribution of the data sets for each stage; for *STS-12*, we reduce the original dataset to guarantee a distribution similar in training and validation dataset concerning WEBNLG. Finally, we split the original data set using 10% for validation.

3.2 Pretrained Language Models

Creating an intermediate vector representation for text and triples requires significant training data to train models from scratch. Limited data is available for alignment, and triples are scarce.

Bringing triple representations to the same-dimensional spaces for text might improve knowledge representation, creating a bridge between KG elements and natural language texts.

We analyze the effects and suitability of pretrained

Table 1: Summary of related studies and their characteristics.

	Text-triple alignment	Pretrained from scratch	Unsupervised	Supervised	NLG	Semantic Representation
Abhishek *et al.* (Abhishek et al., 2022) Regino *et al.* (Regino et al., 2023)	✓			✓	✓	
Lapalme *et al.* (Lapalme, 2020)	✓				✓	
Daw *et al.* (Daw et al., 2021) Pahuja *et al.* (Pahuja et al., 2021)	✓		✓			
Zhu *et al.* (Zhu et al., 2023)		✓	✓			✓
Ours - This work	✓			✓		✓

Table 2: Dataset distribution for fine-tuned model into the siamese networks.

	Train/val	Test
STS-12	2,234	3,108
WebNLG	3,598	774

state-of-the-art encoder models; we fine-tune these models for multimodality data (triple, text) to construct bimodal dimensional spaces. In the following, we present the models chosen and the rationale for our decisions.

E5 encoder. E5 (Wang et al., 2024) refers to a family of encoders trained with contrastive pretrained and weak supervision using approximately a billion text pairs from multilingual datasets from sources such as Wikipedia, Reddit, and others. These E5 models pass through a supervised fine-tuned with high-quality annotated datasets; the knowledge destination technique is used to improve the quality of the embeddings from this family. In particular, we selected the *e5-multi-base* (me5-base) and the *e5-multi-small* (me5-small) model, which map text to 512 and 318 dimension vectors, respectively. We chose e5 because they were trained on curated datasets, which have undergone a rigorous process. This makes them more stable than options trained using unsupervised learning.

DistilUSE.[1] This model (Reimers and Gurevych, 2019) refers to a Multilingual knowledge distilled version of Multilingual Universal Sentence Encoder (Yang et al., 2020) supporting more than 50 languages a sentence and map text into a 512 dimensional. This model represents a light version of bigger models, allowing it to achieve good scores in STS benchmarks.

Paraphrase. This is another model for fine-tuning semantic textual similarity mapping text into 378-dimensional vectors (Reimers and Gurevych, 2019). These models perform well in the semantic similarity task (STS) on the MTEB benchmark (Muennighoff et al., 2023). This is why we took it as a model in our study.

3.3 Data Augmentation Techniques

Our STS-12 dataset focuses on the semantic similarity task in which text only covers text-text alignment. Meanwhile, WEBNLG only covers a triplet-to-text direction since this one contains similarity samples (label 1). Further examples must be used to teach the models when the triples and text differ. Figure 1 presents the ties between triples and text data. The green lines represent the existing relations between data in our dataset; to overcome the unexisting relations (red dashed lines), we proposed a data augmentation technique to increase our dataset.

Triple-Triples (Negative): In contrast with the text, each triple represents unique knowledge. Our technique chooses a triple and randomly selects another one as a negative example to create negative examples of a pair of triples. Additionally, to improve the quality of our augmentation, the triple randomly chosen belongs to the same category. We guarantee this using the category metadata from the datasets.

Triple-Text (Negative): using the same principle that triples are unique (they represent a unique knowledge); and in WEBNLG, such triples are aligned to a unique text (we guarantee this during the preprocessing assigning a unique text to a triple); Our technique selects a random sentence from the WEBNLG data set to align triple with a negative sentence that does not share semantic similarity.

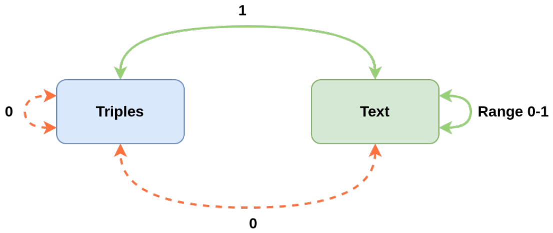

Figure 1: Existing relations between our bimodal datasets - green lines refer to existing relations in our dataset triple-text (WEBNLG) and Text-Text (STS). The orange dashed lines represent the unexisting relations present in our datasets. Label 1 describes a perfect similarity between elements, while 0 indicates no similarity.

[1] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

3.4 Finetune Procedure for Aligning Triples and Text

To finetune our semantic alignment for triples and text, we propose adapting bi-encoder strategies from sentence transformers[2] because their strategy for semantic textual similarity is state-of-the-art (Reimers and Gurevych, 2019), allowing the alignment of the embedding of texts across different languages. In this sense, we select this architecture to align semantic over bimodal data (triple-text).

Figure 2 presents our fine-tuned approach using an adapting bi-encoder network that contains two identical models. Each creates an embedding representation for the left and the right sides *u* and *v*, respectively. Then, in the training strategies, similarity functions such as cosine are used to measure the similarity between the embeddings.

To perform the alignment, we pass the described training dataset from *WEBNLG*, which acts as triple-text pair alignments. Feeding the model only with this dataset could overfit the modeling, losing the semantic learning by the pretrained models. To overcome this problem, we use the *STS-12* dataset during the finetuning to maintain effectiveness in tasks of semantic textual similarity. This allows a common representation between text and triples while still being capable of using the vector from text and creating a representation for triples.

Additionally, as encoder models treat the input as a sequence of characters, we aim to create a semantic embedding from the feeding dataset. We changed the model structure by adding two unique tokens to the vocabulary [TEXT] and [TRIPLES]. Such tokens surround the specific data to characterize better and allow flexible identification.

We fine-tuned the model using 90% of training and 10% for validation. We select the same distribution for each dataset (*WEBNLG* and *STS-12*); as our main objective is to align triples and text, we use an embedding Information Retrieval evaluator from sentence transformers to assess the capacity of the models to recover other kinds of data and decide whether to continue training and save the best models.

Triples and textual data represent knowledge using structured and unstructured data, respectively. We decided to test our study regarding supervised learning with two different losses to identify the more suitable one.

Contrastive Loss. (Hadsell et al., 2006) This is a loss that does not follow a similarity score (range of similarity). To compute the loss, contrastive requires a pair of sentences: the first one is called anchor, and this one is linked to another sentence that can be a positive example (similar example to the anchor represented with label 1) and negative example (example with dissimilarity with label 0). For this scenario, all triple-text pairs are linked with a positive label. In *STS-12* dataset with a filter, sentences with high similarity values greater than 0.9 are given label one, while the remaining data are labeled zero.

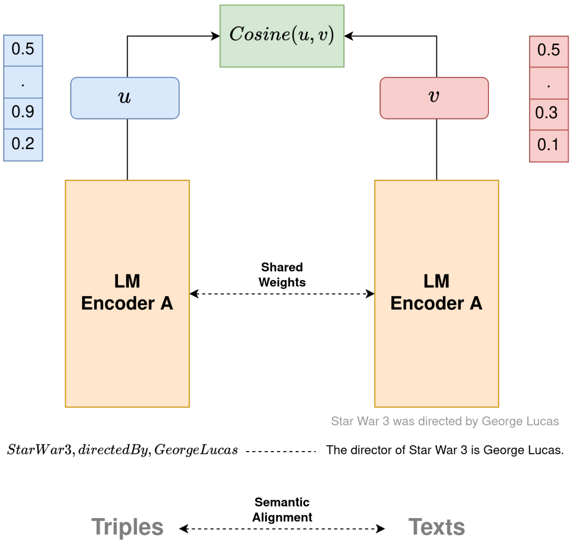
0.5
.
0.9
0.2
$Cosine(u,v)$
u
v
0.5
.
0.3
0.1
LM Encoder A
Shared Weights
LM Encoder A
Star War 3 was directed by George Lucas
$StarWar3, directedBy, GeorgeLucas$
The director of Star War 3 is George Lucas.
Triples
Semantic Alignment
Texts

Figure 2: Fine-tuning a bi-encoder to align triples and texts.

Multiple Negatives Symmetric Ranking Loss(Henderson et al., 2017) This is a loss function that only considers a pair of sentences: an anchor and a positive (similar sentence to the anchor); this loss computes first the loss for finding the positive for a given anchor, and then, finding the anchor for a given positive example.

Algorithm 1 presents our developed finetuning process using different loss functions: (1) contrastive, (2) contrastive with data augmentation, and (3) Multiple Negative Symmetric Ranking Loss (MNSRL); for the latter, we do not apply data augmentation due to only need positive examples.

First, we pass four pre-trained models (line 1). Each pretrained Encoder is selected to start the finetuned; then, we add the datasets *STS-12* and *WEBNLG* (line 2); afterward, we chose each loss function configuration (line3); depending on the loss, we conducted intermediate steps to filter the adequate data (line 4 and 8). For example, we increased the dataset with our data augmentation or filtered only the positive examples for Multiple Negative Symmetric Ranking Losses. Finally, the process begins with the finetuned information retrieval task, passing the train data, the pretrained encoder, and the loss (line 10).

[2] `https://www.sbert.net`

Algorithm 1: Iterative Fine-tuned process for text triples alignment using Constrative and MultipleNegativesSymmetricRanking loss (MNSRL).

```
Input: $DistilUSE, Paraphrase, e5-base, e5-small$
Data: $STS12, WEBNLG$
Result: 12 Finetuned Text-Triples $FineEncoder_{lossEncoder}$ for each model configuration
foreach $(Encoder) \in DistilUSE, Paraphrase, e5-base, e5-small$ do
    $TrainData \leftarrow joinData(STS-12, WEBNLG)$
    foreach $(loss) \in Constrative, ConstrativeAug, MNSRL$ do
        if $loss == ConstrativeAug$ then
            $TrainData \leftarrow DA(TrainData)$
        end
        if $loss == MSRL$ then
            $TrainData \leftarrow FilterPositive(TrainData)$
        end
        $FineEncoder_{lossEncoder} \leftarrow Finetuned_{STS}(TrainData, Encoder, loss)$
    end
end
```

4 EVALUATION METHODOLOGY

This section presents the evaluation methodology for the fine-tuned models created via our approach. Our solution creates embedding for text and triples, and this study evaluates them using two main tasks. Figure 3 presents the evaluation methodology using the two primary datasets explored in this study. Their use emphasizes validating the models' effectiveness over specific tasks described in the following sections. We compare our results of fine-tuned models with the original models without any tuning. In the following, we explain each task and metrics.

4.1 Task 1: Sentence Similarity

The semantic similarity task consists of computing the similarity between the encoder representations of two sentences and comparing the results with a human label score of similarity (measuring the correlation).

Figure 4 presents how we perform our evaluation. First, we extract samples from the STS-12-Test partition; as each sample is composed of a pair of sentences with a score of similarity, then we used the encoder (original pre-trained or our generated fine-tuned ones) to encode sentences (S_1 and S_2). Afterward, we compute the similarity between the encoded vectors ($V1_i$ and $V2_i$) using the cosine function. Finally, we compare *CosineScore* with *score* and compute the correlation. This shows whether the models' embedding performs well with human judgments to score a grade of similarity.

4.2 Task 2: Information Retrieval

This evaluation aims to assess the system's capacity to retrieve the alignment for a particular query correctly. For example, when presented with a "triple", the system should provide the associated text that accurately represents its semantic meaning. Conversely, when given an input text query, the system should retrieve the appropriate triple. Figure 5 presents the two key steps: (A) the Population step, which creates the vectors from the test datasets, and (B) the semantic search step to recover the one with the highest similarity.

In this evaluation, we start from an alignment (X,Y); we conduct two main paths, one in X being a triple and Y as a text; this coverage is a direction ($triples \rightarrow text$). After that, we replicate the steps having X as text and Y as triple to cover the ($text \rightarrow triples$ direction).

We describe each step for validating the information retrieval task in the following.

A) Population

Figure 5-A presents the process for our alignment evaluation for IR-Task. First, we extract text from the WEBNLG dataset in the test portion. Each of the sentences is associated with unique triples. We pass X (Triple or sentences) through our encoder models (zero-shot or fine-tuned). This creates a vector for each X (X_{vector}), and then we save each vector in a database to enable the semantic search.

B) Semantic Search

Figure 5-B presents the evaluation with information retrieval in which a given Y_i (triple or text depending on the direction of the evaluation) from WEBNLG; we represent into a vector form using our encoder, then the vector passes to the semantic search module in which this vector recovers the sentence with higher similarity using cosine metrics to compare the query vector with the vectors stored in our database.

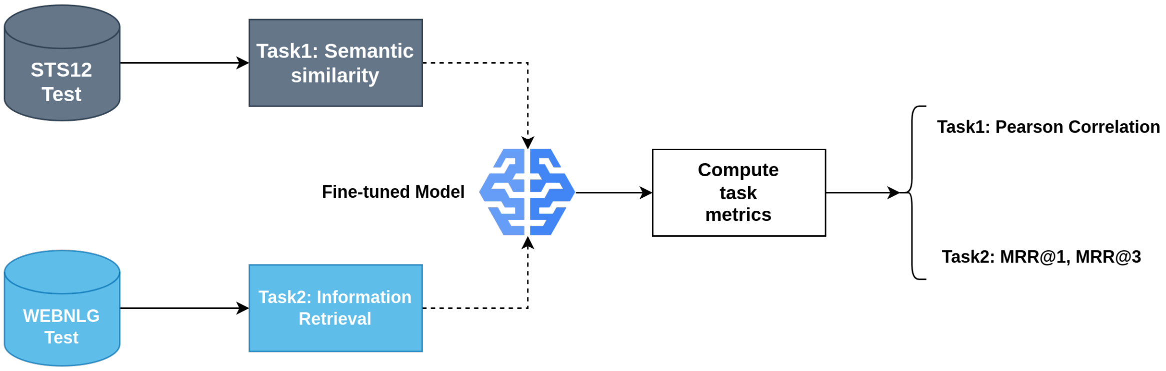
STS12 Test
Task1: Semantic similarity
Fine-tuned Model
WEBNLG Test
Task2: Information Retrieval
Compute task metrics
Task1: Pearson Correlation
Task2: MRR@1, MRR@3

Figure 3: Overall evaluation procedure regarding the generated encoders through two main tasks: Semantic similarity and Information Retrieval. Specific metrics are computed for these tasks.

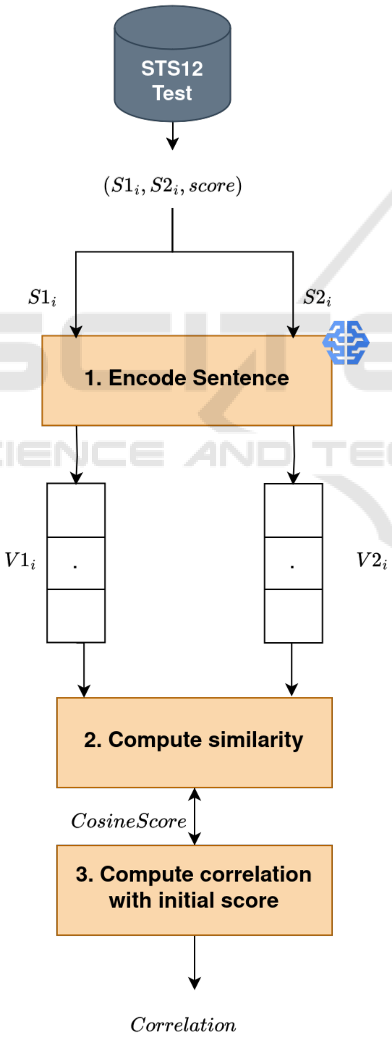

Figure 4: Evaluation for Task 1: Sentence Similarity.

4.3 Evaluation Metrics

We selected the most suitable metrics for each of the investigated evaluation tasks.

Pearson Correlation Coefficient - Task 1

We choose Pearson's correlation coefficient for STS, which evaluates the linear correlation between two datasets. This enables us to compare the human annotation score with the cosine similarity of the embeddings of the trained models. The metrics values take values between -1 and +1. The higher values represent a higher correlation.

Mean Reciprocal Rank (MRR) - Task 2

We approach the alignment problem as a retrieval issue, opting to use Mean Reciprocal Rank (MRR) as our primary metric for evaluating the accuracy of the alignment between text and triples. We chose MRR because it is a ranking metric that considers the position of the correct candidate when assigning scores. This benefits our needs because it prioritizes higher scores for correctly identifying the top candidates. In conclusion, we decided to utilize MRR@1 (indicating whether the system identified the right candidate) and MRR@3 (to determine if the system correctly identified the right candidate among the top 3 recovered documents) as our evaluation criteria.

5 RESULTS

This section presents the experimental results of evaluating the pretrained models (without fine-tuning) and applying them to fine-tune based on our developed approaches using two different loss functions and our data augmentation technique. We show the

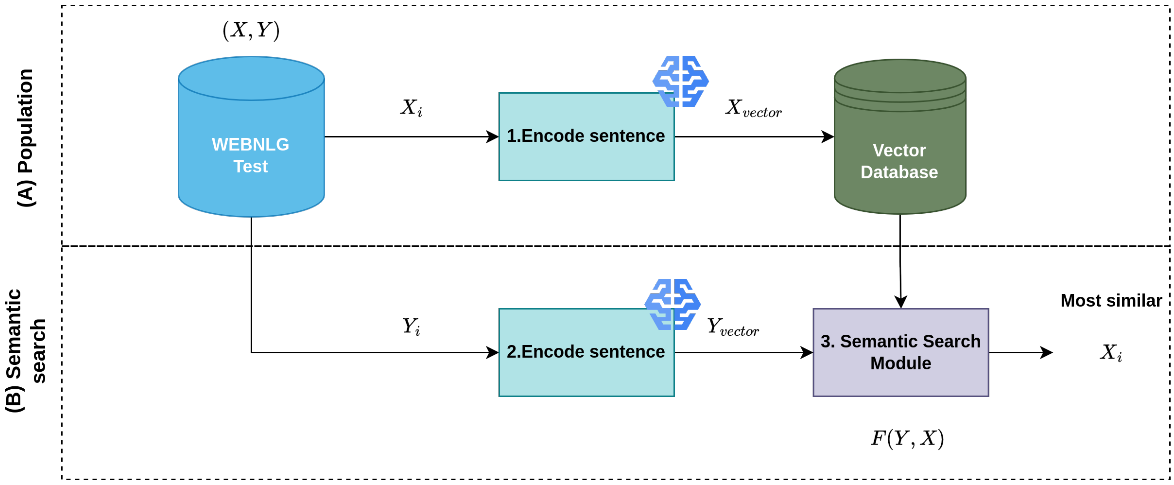

Figure 5: Evaluation concerning Task 2: Information Retrieval; A) Populating the triples from WEBNLG dataset to a vector store database; and B) Semantic search evaluation to evaluate the models in the Information Retrieval task.

results for each evaluation task: Sentence Similarity (cf. Subsection 5.1) and Information Retrieval (cf. Subsection 5.2).

5.1 Result - Task 1: Sentence Similarity

Table 3 presents the results of STS-12 test datasets of the fine-tuned models. We note that for the original pre-trained models, the paraphrased models achieved better correlation, followed by models from the e5 family. For the finetuned models with Contrastive loss, only distil-use increases the value of the Pearson coefficient, while the others decrease their values.

Applying data augmentation, we noticed a gain in Pearson with e5-small and paraphrase. The other models' Pearson coefficient values decreased, but their values did not deviate significantly from the original.

Finally, MNSRL achieved better results with the multilingual e5 base and increased effectiveness for almost every model except for distil use.

Table 3: Results for Similarity Evaluator using STS-12 test dataset and Pearson correlation to measure Coherence with the human annotation. Over models without fine-tuned (original), Contrastive Loss (LS) without and with Data augmentation and using Multiple Negatives Ranking Loss (MNSRL).

	STS-12 Pearson Correlation Test set			
Models	**Original**	**CL.**	**CL. DA**	**MNSRL**
me5-base	0.8426	0.8167	0.8358	**0.8535**
me5-small	0.8416	0.8152	**0.8523**	0.8373
distil-use	0.7935	**0.8275**	0.7907	0.8246
Paraphrase	0.8470	0.8275	**0.8533**	0.8510

5.2 Result - Task 2: Information Retrieval

Task 2: Information Retrieval Text-Triples

Table 4 presents the results in terms of MRR@1 and MRR@3 for IR using the embedding of text to recover triples; the pretrained models (original) over e5-small achieved the highest values among pre-trained models 0.6602 and 0.7965 for MRR@1 and MRR@3, respectively.

Fine-tuned models with contrastive loss positively affect the paraphrase multilingual models, while the others suffer degradations of MRR@1. Moreover, applying data augmentation techniques looks like the degradation persists in terms of MRR@1, but we also notice a slight growth in terms of MRR@3 (paraphrase and distill-use)

We observed that fine-tuned models with MNSRL slightly improved over all the models in MRR@1 and MRR@3, which have notorious effects on the paraphrase model.

Task 2: Information Retrieval Triples-Text

Table 5 presents the results of given a query triple retrieval similar texts over this table. Pretrained models perform slightly lower than text-triple (cf. Table 4).

Similarly to the last table, the pre-trained models from the e5-family achieved the best results without any fine-tuning.

The e5 models reduce their MRR scores for contrastive loss, while the others increase the MRR score values. We also noticed that data augmentation produces positive effects compared to the CL finetuned

Table 4: Mean Reciprocal Rank (MRR@[1,3]) for Information Evaluator over the test dataset from WEBNLG challenge using text as query and triples as a corpus for retrieval. Using Encoder without finetunning (Original), fine-tuned with contrastive loss (CL) without and with Data augmentation (DA) and Multiple Negatives Symmetric Ranking Loss (MNSRL).

	Text as query and Triples as a corpus for retrieval (MRR)							
	MRR@1				MRR@3			
Models	**Original**	**CL.**	**CL. DA**	**MNSRL**	**Original**	**CL.**	**CL. DA**	**MNSRL**
me5-base	**0.6576**	0.6382	0.6499	0.6693	0.7937	0.7786	0.7929	**0.8058**
me5-small	**0.6602**	0.6447	0.6499	0.6654	0.7965	0.7883	0.7907	**0.8019**
distil-use	0.6395	0.6318	**0.6499**	0.6408	0.7816	0.7877	0.7866	**0.7892**
Paraphrase	0.5917	**0.6473**	0.6370	0.6460	0.7334	0.7705	0.7810	**0.8531**

Table 5: Mean Reciprocal Rank (MRR@[1,3]) for Information Evaluator over the test dataset from WEBNLG challenge using triples as query and text as a corpus for retrieval. Using Encoder without finetunning (Original), fine-tuned with contrastive loss (CL) without and with Data augmentation (DA), and Multiple Negatives Symmetric Ranking Loss (MNSRL).

	Triples as query and Text as a corpus for retrieval (MRR)							
	MRR@1				MRR@3			
Models	**Original**	**CL.**	**CL. DA**	**MNSRL**	**Original**	**CL.**	**CL. DA**	**MNRSL**
me5-base	0.6537	0.5969	0.6499	**0.6615**	0.7935	0.7418	0.7892	**0.7991**
me5-small	**0.6576**	0.5736	0.6395	0.6525	**0.7922**	0.7149	0.7784	0.7907
distil-use	0.6279	0.6408	0.6473	**0.6486**	0.7689	0.7780	0.7827	**0.7901**
Paraphrase	0.5646	0.5930	0.6292	**0.6447**	0.7110	0.7330	0.7661	**0.7817**

models. However, this improvement does not reach values higher than pretrained models without tunned.

Finally, when we applied the MNSRL loss function, model e5 achieved the highest MRR (0.6615) out of all the models tested. Additionally, this loss function boosted all the models tested, resulting in similar MRR@3 values.

6 DISCUSSION

The pretrained encoder models achieved reasonable results over the two tested tasks; we notice that some of them, such as e5 flavors, present well without any tuning over task 2; this might explained by the similarity grammar presented between entities and texts.

We observed that the models achieved good overall results in the Task 1, with all Pearson coefficients higher than 0.79 (cf. Table 3), which correlates well with human annotation. The results show that preserving the semantics (STS) was possible in all configurations, showing that the training did not considerably decrease the Pearson correlation score.

We evaluated our models to perform retrieval in two directions: returning the most similar triple given a text query into natural text (Text—Triples, cf. Table 4) and retrieving the most similar text given a triple (Triple—Text, cf. Table 4), where e5-base finetuned with MNSRL achieved the best effectiveness in terms of MRR@1 having 0.6693 and 0.6615 for retrieval text-triples and triples-text, respectively. The models can reasonably recover the associated triple or text.

The paraphrase shows an increase in the MRR metrics (cf. Table 4, Table 5), a comparison with STS results (cf. Table 3) reveals a maintaining in Pearson correlation. This suggests that the observed improvements are learning triples and text alignment while preserving the capability of the models in task 1. Our results for the information retrieval task showed slight improvements. This happens because the WEBNLG test dataset contains some relations and entities not present in the training dataset. As a result, it can be more challenging to accurately identify the connections between the text and the triples to achieve correct alignments.

Additionally, applying the data augmentation technique demonstrated positive effects when used in combination with constructive loss and improved their results in IR Triple-text (Table 5)). This improvement can also be appreciated by focusing on MRR@3 metrics related to the retrieval system having the correct sentence in the tree of most relevant retrieved sentences.

Furthermore, the MNRSL loss was more adequate for this experiment. This evidence shows we can align text and triples by only providing positive examples. The results were comparable and also outperformed our results with data augmentation.

We found that having datasets with both triples and text examples is necessary to accelerate the development of stronger models. Creating these datasets could potentially impact the performance of our tested loss, particularly for MNRSL. We observed that the models maintain semantic integrity in Task 1 while delivering improved results in Task 2.

7 CONCLUSION

Embeddings are crucial in combining KGs and language models in modern digital applications. Nevertheless, the literature lacks investigations into how to create embeddings that align triples and text representations using small-size datasets. This study advanced the state-of-the-art by exploring triple embedding using pretrained models and attaching the evaluation to two relevant tasks. We identified the need to improve the alignment between text and triples before directly applying the pretrained models. Our findings revealed that some models responded well to the finetuning improvement of their MRR@1 score (retrieval task). This suggests that the nature of the models and more refined techniques could contribute to better alignment strategies for triples and texts. We demonstrated that using multiple negatives symmetric ranking loss enables semantic learning using all the pretrained models in a small dataset. Our findings indicate that triple embeddings benefited from data augmentation with Contrastive loss in combination with text-text data (STS). Future research involves creating our triple-text datasets to increase their richness for alignment. We plan to explore other languages besides English to measure whether the applications of low-resource languages are applicable. Finally, we intend to examine the potential for self-supervised learning techniques in enhancing text-triples' alignment.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. This work is also supported by the 'PIND/FAEPEX - "Programa de Incentivo a Novos Docentes da Unicamp" (#2560/23) and the São Paulo Research Foundation (FAPESP) (Grant #2022/15816-5)[3]

[3] The opinions expressed in this work do not necessarily reflect those of the funding agencies.

REFERENCES

Abhishek, T., Sagare, S., Singh, B., Sharma, A., Gupta, M., and Varma, V. (2022). Xalign: Cross-lingual fact-to-text alignment and generation for low-resource languages. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 171–175, New York, NY, USA. Association for Computing Machinery.

Cao, J., Fang, J., Meng, Z., and Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Comput. Surv.*, 56(6).

Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., and Shimorina, A. (2020). The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., and Shimorina, A., editors, *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Chang, T. A. and Bergen, B. K. (2024). Language Model Behavior: A Comprehensive Survey. *Computational Linguistics*, pages 1–58.

Daw, S., Sagare, S., Abhishek, T., Pudi, V., and Varma, V. (2021). Cross-lingual alignment of knowledge graph triples with sentences. In Bandyopadhyay, S., Devi, S. L., and Bhattacharyya, P., editors, *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 629–637, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ding, L., Kolari, P., Ding, Z., and Avancha, S. (2007). *Using Ontologies in the Semantic Web: A Survey*, pages 79–113. Springer US, Boston, MA.

Fionda, V. and Pirrò, G. (2020). Learning triple embeddings from knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3874–3881.

Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). Creating training corpora for NLG micro-planners. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Henderson, M., Al-Rfou, R., Strope, B., hsuan Sung, Y., Lukacs, L., Guo, R., Kumar, S., Miklos, B., and Kurzweil, R. (2017). Efficient natural language response suggestion for smart reply.

Hogan, A. (2020). *SPARQL Query Language*, pages 323–448. Springer International Publishing, Cham.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. (2022). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Kalinowski, A. and An, Y. (2022). Repurposing knowledge graph embeddings for triple representation via weak supervision. In *2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pages 129–137.

Lapalme, G. (2020). RDFjsRealB: a symbolic approach for generating text from RDF triples. In Castro Ferreira, T., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., and Shimorina, A., editors, *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 144–153, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2).

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2023). MTEB: Massive text embedding benchmark. In Vlachos, A. and Augenstein, I., editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Pahuja, V., Gu, Y., Chen, W., Bahrami, M., Liu, L., Chen, W.-P., and Su, Y. (2021). A systematic investigation of KB-text embedding alignment at scale. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1764–1774, Online. Association for Computational Linguistics.

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20.

Patil, R., Boit, S., Gudivada, V., and Nandigam, J. (2023). A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11:36120–36146.

Perković, G., Drobnjak, A., and Botički, I. (2024). Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pages 2084–2088.

Qader, W. A., Ameen, M. M., and Ahmed, B. I. (2019). An overview of bag of words;importance, implementation, applications, and challenges. In *2019 International Engineering Conference (IEC)*, pages 200–204.

Regino, A. G., Caus, R. O., Hochgreb, V., and dos Reis, J. C. (2023). From natural language texts to rdf triples: A novel approach to generating e-commerce knowledge graphs. In Coenen, F., Fred, A., Aveiro, D., Dietz, J., Bernardino, J., Masciari, E., and Filipe, J., editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 149–174, Cham. Springer Nature Switzerland.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., and Wei, F. (2024). Text embeddings by weakly-supervised contrastive pre-training.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Xia, P., Wu, S., and Van Durme, B. (2020). Which *BERT? A survey organizing contextualized encoders. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7516–7533, Online. Association for Computational Linguistics.

Yan, Q., Fan, J., Li, M., Qu, G., and Xiao, Y. (2022). A survey on knowledge graph embedding. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*, pages 576–583.

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Hernandez Abrego, G., Yuan, S., Tar, C., Sung, Y.-h., Strope, B., and Kurzweil, R. (2020). Multilingual universal sentence encoder for semantic retrieval. In Celikyilmaz, A. and Wen, T.-H., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Zhu, H., Peng, H., Lyu, Z., Hou, L., Li, J., and Xiao, J. (2023). Pre-training language model incorporating domain-specific heterogeneous knowledge into a unified representation. *Expert Systems with Applications*, 215:119369.

Zhu, Y., Wan, J., Zhou, Z., Chen, L., Qiu, L., Zhang, W., Jiang, X., and Yu, Y. (2019). Triple-to-text: Converting rdf triples into high-quality natural languages via optimizing an inverse kl divergence. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 455–464, New York, NY, USA. Association for Computing Machinery.