GEM: Translation-Free Zero-Shot Global Entity Matcher for Global Catalogs

Karim Bouyarmane
bouykari@amazon.com
Amazon
Seattle, Washington, USA

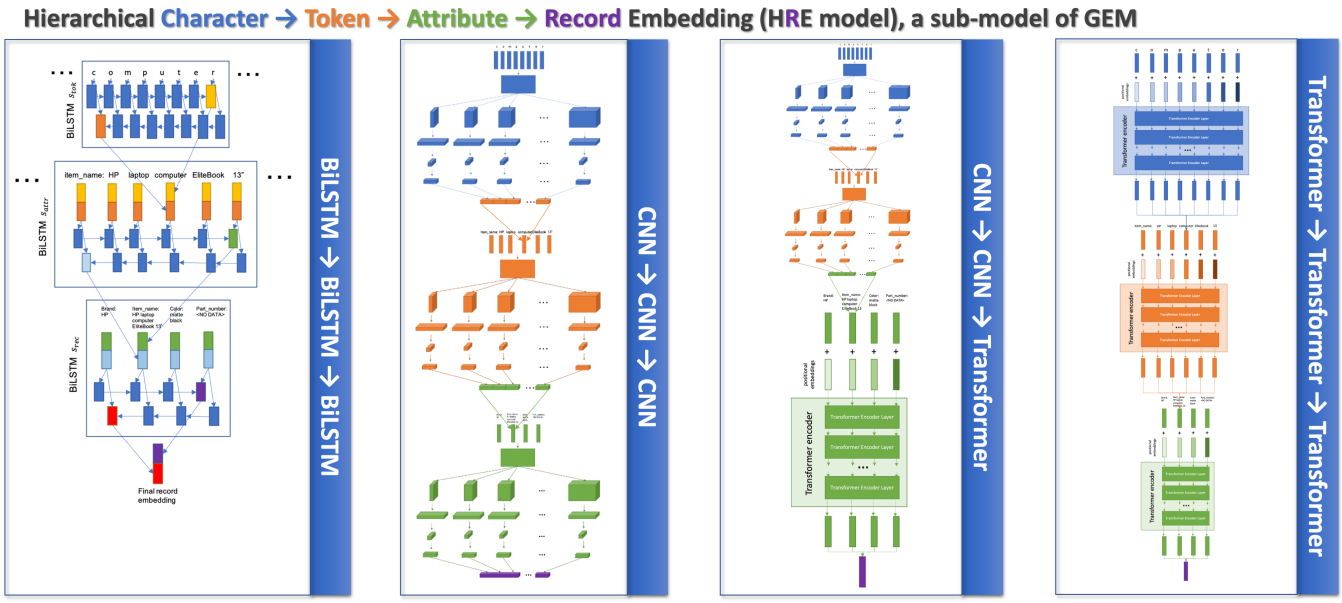

Figure 1: Various implementations of Hierarchical Record Embedding models (HRE) with different sequence embeddings.

ABSTRACT

We propose a modular BiLSTM / CNN / Transformer deep-learning encoder architecture, together with a data synthesis and training approach, to solve the problem of matching catalog products across different languages, different local catalogs, and different catalog data contributors. The end-to-end model relies solely on raw natural language textual data in the catalog entries and on images of the products, without any feature engineering, and is entirely translation-free, not requiring the translation of the catalog natural-language data to the same base language for inference. We report experiments results on a 4-languages-scope model (English, French, German, Spanish) matching entities from 4 local catalogs (UK, France, Germany, Spain) of a retail website. We demonstrate that the model achieves performance comparable to state-of-the-art existing entity matchers that operate within a single language, and that the model achieves high-performance zero-shot inference on language pairs not seen in training.

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08.
https://doi.org/10.1145/3447548.3467209

CCS CONCEPTS

• **Information systems** → **E-commerce infrastructure**; **Online shopping**; • **Computing methodologies** → **Machine learning algorithms**; **Neural networks**; **Natural language processing**.

KEYWORDS

Hierarchical neural networks; record embedding; record representation learning; record matching; cross-lingual matching; deduplication

ACM Reference Format:
Karim Bouyarmane. 2021. GEM: Translation-Free Zero-Shot Global Entity Matcher for Global Catalogs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3447548.3467209

1 INTRODUCTION

1.1 The Entity Matching Problem

Entity Matching (EM), also known as *Entity Resolution*, *Duplicate Detection*, *Record Matching*, *Record Linkage*, consists in identifying

the different manifestations of the same real-world entity in one or multiple databases of records/listings/entries [4, 6].

In this paper, we are interested in the instance of the problem applied to shopping websites. Given two product listings and their associated information, we want to decide wether they are two different listings of the same product. This can occur for example when multiple sellers sell the same product on the website.

State-of-the-art Entity Matching models rely on both the textual data in the input pair of records (structured in *attributes* or *fields*) and on the images of the products. Seen as a natural language corpus, the characteristics of the textual data in the attributes are **1**. a very high occurrence rate of named entities and codes, **2**. domain-specific vocabulary, and **3**. a mix of sequentially-structured natural language and bags-of-words. Image data alone, when present, is not sufficient to match entities. Finally, the data in the different fields can be missing or very noisy.

These challenges are inherent to the scale of operations of the largest shopping websites, and they translate into significant challenges for the EM problem. Existing EM models use hand-crafted matching features across selected pairs of attributes of the two products, random-forest or boosted decision-trees classifiers on the features, and more recently deep-learning models and classifiers on the features [3, 5, 12, 13, 16].

The input features of these models are textual features of the products (TF-IDF features, pre-trained word embeddings), and pairwise matching features, notably string similarity features between pairs of attributes in the products (e.g. Jaccard similarity, Levenshtein distance, etc.) [2].

1.2 Global Entity Matching

Global Entity Matching (GEM) is the problem of matching entities across stores in different geographies and different languages.

To the best of our knowledge, no existing work has addressed Global Entity Matching in the literature.

The GEM problem adds new challenges to classical EM problem as discussed in Sec. 1.1. The first and main one is matching entities across language borders. Existing EM approaches based on string-similarity features and attribute-pairs-matching features do not apply to the problem. E.g. Record 1: {title: "iPhone X", color: "black"} and Record 2: {title: "iPhone X noir", color: <empty field>} would produce a no-match using existing EM models if the model does not know that Record 1 is in English and Record 2 is in French and that *noir* in French means *black* in English. A second added challenge to vanilla EM is the possible schema mis-alignment across the local catalogs. Each local store might be using a local catalog schema slightly different from the other ones due to local regulatory or cultural specificities.

One naive, brute-force, approach is to use Machine Translation (MT) to translate the two entities in the pair to the same base language then use the existing same-language EM models on the translated pairs.

This approach suffers from several drawbacks. The first one is the scalability and cost, as it requires to translate hundreds of millions of products data per day. The second problem is that general MT systems, trained on general natural-language parallel corpora, tend to produce low-quality product-data translations,

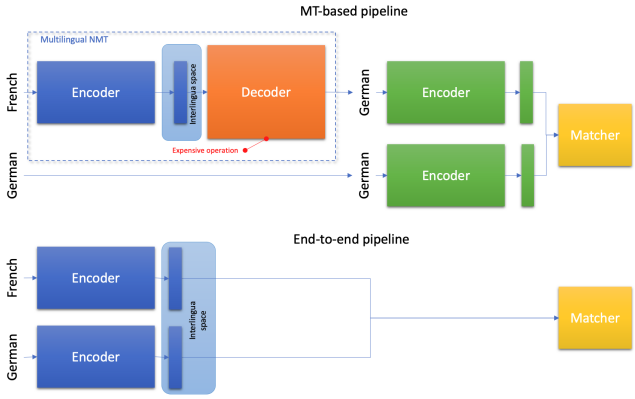

Figure 2: MT-based pipeline vs end-to-end pipeline.

as the language model of the catalog data differs substantially from a general-domain language model. Finally, using state-of-the-art seq2seq Neural Machine Translation (NMT) systems [7, 9, 15] consisting in an encoding step and decoding step requires more effort and data than strictly necessary for the problem at hand by adding a redundant step in the pipeline. Indeed, after the decoding step, the resulting translation would need to be re-encoded to an embedding space where a matching metric or similarity metric is to be learned, thus resulting in a redundant [encoding → decoding → encoding → matching] pipeline instead of only a [encoding → matching] pipeline. See Fig. 2.

We propose an alternative, translation-free, end-to-end approach that directly matches records across languages. The difficulty with such an end-to-end approach, however, is that it would require the collection of millions of manually-labeled training pairs that are slow and costly to obtain. GEM being a new problem, there is no available labeled dataset that can be readily used for training, unlike the vanilla EM datasets. Therefore, we propose to leverage the existing same-language data only and synthesize from it the necessary dataset to train the GEM model.

The core component of the model is a modular Hierarchical Record Embedding deep-learning encoder, used to encode the entities to a language-agnostic representation (a "universal interlingua") before matching them. This is an extension of the concept of language-agnostic representations (or aligned representations) of words [8] or sentences [19] to records. Similarly to Multilingual NMT systems [9, 15], we train the model on rotations of language pairs from the synthetic dataset to force the language-agnosticity and universality of the encoder. We demonstrate that this allows to accurately match entities across pairs of languages unseen in the synthetic training dataset.

2 PROBLEM FORMULATION

2.1 Hierarchical Record Embedding

In this section we define a general character-level embedding model for records in the catalog. Compared to other record embedding approaches [1, 20], we chose to operate at character-level as character vocabularies are mostly agnostic to the languages (or, at least, the

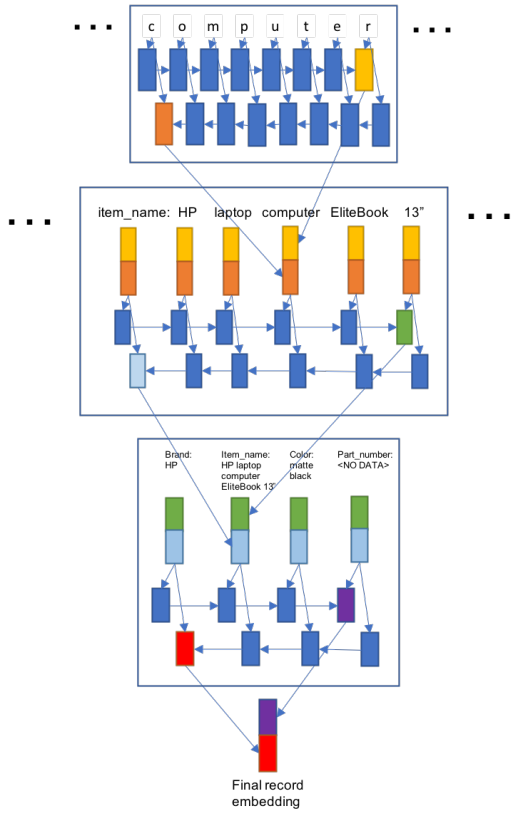

Figure 3: Example instantiation of hierarchical record embedding with BiLSTM.

union of character vocabularies across most languages remains tractable, as opposed to the union of word vocabularies. This excludes languages like Chinese which are out of the scope of this work). Character-level models also help overcome the problem of a high-rate of out-of-vocabulary words [14], as well as a high-rate of rare named-entities, that characterize large e-commerce catalogs. They also help overcome the problems of correctness of the data (for example to be robust against spelling mistakes in a *model number* attribute and to correctly interpret the prefixes and suffixes of several kinds of codes and identifiers). Finally, character-level models are robust against mixing languages within a single record (*brand* in English while the rest of the text is in French for example).

We model a record as a 3-level hierarchical sequence, down from the character level and up to the record level. A *character* is any unicode character that appears in a predefined set of most-frequent characters in the combined character corpora across all local catalogs. We keep this set small (at 100 characters), with all characters out of this set replaced with a unique OOV character. A *token* (or *word*) is a sequence of *characters*. An *attribute* (for example, the *title* attribute, the *color* attribute, etc.) is a sequence of *tokens*. And finally the *record* itself is a sequence of *attributes*. Usually, records are fixed-length sequences of attributes (same set of attributes for all the records corresponding to the schema of the catalog, although some attributes might be empty, we keep them as empty strings). However, to keep the approach general and robust against local catalogs schema misalignments, we can also consider the records as variable-length sequences of attributes. We additionally prepend the attribute value with the name of the attribute itself. For example, if the attribute *title* is the string "iPhone 10 space grey" we transform it to become the string "title: iPhone 10 space grey". No other pre-processing is done on the raw data. Attributes are variable-length sequences of tokens. Tokens are variable-length sequences of characters.

Let us denote $\mathcal{S}$ the set of all real-world entities (or products). A catalog $\mathcal{C}$ is a set of N records r_i, $i \in \{1, \ldots, N\}$. The catalog $\mathcal{C}$ is a subset of the set of all possible records $\mathcal{V}$. On $\mathcal{V}$ we define the equivalence relation $\sim$ as $r_1 \sim r_2$ if r_1 and r_2 describe the same real-word entity e.

A record $r = (a_1, \ldots, a_n)$ is an n-tuple of n attributes, each attribute $a_i = (t_{i1}, \ldots, t_{im_i})$ is a variable-length sequence of m_i tokens (words). Each token $t_{ij} = (c_{ij1}, \ldots, c_{ijl_{ij}})$ is a variable-length sequence of l_{ij} characters. Each character c_{ijk} is an element of $V = \{1, \ldots, v\}$ where v is the total number of characters in the character vocabulary of the corpus.

We define a maximum number of characters per token λ and a maximum number of tokens per attribute μ. For a token $t_{ij} = (c_{ij1}, \ldots, c_{ijl_{ij}})$, the equalized-length token $\bar{t}_{ij}$ is the token $\bar{t}_{ij} = (\bar{c}_{ij1}, \ldots, \bar{c}_{ij\lambda})$ where

$$\bar{c}_{ijk} = \begin{cases} c_{ijk} & \text{if } k \leq l_{ij}, \\ 0 & \text{if } k > l_{ij}. \end{cases} \tag{1}$$

Similarly, for an attribute $a_i = (t_{i1}, \ldots, t_{im_i})$, the equalized-length attribute $\bar{a}_i$ is the attribute $\bar{a}_i = (\underline{t}_{i1}, \ldots, \underline{t}_{i\mu})$ where

$$\underline{t}_{ij} = \begin{cases} \bar{t}_{ij} & \text{if } j \leq m_i, \\ \varnothing & \text{if } j > m_i. \end{cases} \tag{2}$$

For a record $r = (a_1, \ldots, a_n)$, the equalized-length record $\bar{r} = (\bar{a}_1, \ldots, \bar{a}_n)$ is thus an order 3 tensor of shape $n \times \mu \times \lambda$ in $\mathcal{V} = \left(\left(\bar{V}^{\lambda}\right)^{\mu}\right)^{n}$ where $\bar{V} = V \cup \{0\}$.

The catalog $\mathcal{C}$ can now be seen as a subset of $\mathcal{V}$ (i.e. $\mathcal{C} \subset \mathcal{V}$) and the problem of Entity Matching in $\mathcal{V}$ consists in learning the equivalence relation $\sim$ on $\mathcal{V}$. The quotient space $\mathcal{V}_{/\sim}$ is the set of all real-world entities, i.e. $\mathcal{S} \equiv \mathcal{V}_{/\sim}$.

Definition 1. A sequence embedding model s from a space of dimension p to a space of dimension q is a mapping

$$s : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^p\right)^i \to \mathbb{R}^q \tag{3}$$

that maps every ordered finite sequence of elements of $\mathbb{R}^p$ to exactly one element of $\mathbb{R}^q$.

Several deep-learning architectures have been proposed in the literature to implement a sequence embedding model. In this paper we experimented with three of them: BiLSTM [19], sequence-CNN (seqCNN) [10], and Self-Attention Transformer [21].

In the remaining developments of this section we will use the following definition of the operator map (higher-order function):

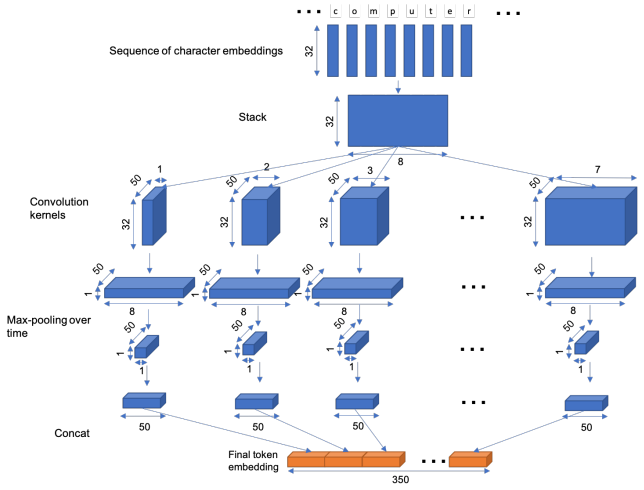

Figure 4: a SeqCNN s_{tok}.

Definition 2. for a function $f : A \to B$, we define $\text{map}(f)$ as

$$\begin{aligned} \text{map}(f) : \bigcup_{i=0}^{\infty} A^i &\longrightarrow \bigcup_{i=0}^{\infty} B^i \\ (\alpha_1, \ldots, \alpha_k) &\longmapsto (f(\alpha_1), \ldots, f(\alpha_k)) \,. \end{aligned} \tag{4}$$

We can now define the notion of hierarchical embedding model.

Definition 3. The hierarchical composition of two sequence embedding models $s_1 : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{p_1}\right)^i \to \mathbb{R}^{p_2}$ and $s_2 : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{p_2}\right)^i \to \mathbb{R}^{p_3}$, denoted $\text{comp}(s_1, s_2)$, is the function

$$h : \bigcup_{i_2=0}^{\infty} \left(\bigcup_{i_1=0}^{\infty} \left(\mathbb{R}^{p_1}\right)^{i_1} \right)^{i_2} \longrightarrow \mathbb{R}^{p_3} \tag{5}$$

defined as

$$h = \text{comp}(s1, s2) = s_2 \circ \text{map}(s_1) \,. \tag{6}$$

Two-level hierarchical embedding models have been implicitly used, e.g. in [11, 14]. More generally, we define hierarchical embeding models for a n-level hierarchy of n sequence embedding models:

Definition 4. The hierarchical composition of n sequence embedding models $s_1 : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{p_1}\right)^i \to \mathbb{R}^{p_2}$, $s_2 : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{p_2}\right)^i \to \mathbb{R}^{p_3}$, $\ldots$, $s_n : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{p_n}\right)^i \to \mathbb{R}^{p_{n+1}}$ is the function

$$h : \bigcup_{i_n=0}^{\infty} \left(\bigcup_{i_{n-1}=0}^{\infty} \left(\ldots \left(\bigcup_{i_1=0}^{\infty} \left(\mathbb{R}^{p_1}\right)^{i_1} \right)^{\iddots} \right)^{i_{n-1}} \right)^{i_n} \longrightarrow \mathbb{R}^{p_{n+1}} \tag{7}$$

defined as

$$h = \text{comp}(s_1, s_2, \ldots, s_n) \tag{8}$$

$$= s_n \circ \text{map}\left(s_{n-1} \circ \text{map}\left(\ldots \circ \text{map}\left(s_2 \circ \text{map}(s_1) \right) \right) \right). \tag{9}$$

Let us define a character embedding layer

$$e_{\text{char}} : \bar{V} \to \mathbb{R}^{d_{\text{char}}} \tag{10}$$

and its transformation $\text{map}(\text{map}(\text{map}(e_{\text{char}})))$:

$$\text{map}^3(e_{\text{char}}) : \mathcal{V} = \left(\left(\bar{V}^{\lambda} \right)^{\mu} \right)^{n} \longrightarrow \mathcal{R} = \left(\left(\left(\mathbb{R}^{d_{\text{char}}} \right)^{\lambda} \right)^{\mu} \right)^{n} \tag{11}$$

which embeds the records as order-4 real tensors from their initial representation as order-3 integer tensors [17]. We define three sequence embedding models:

$$s_{\text{tok}} : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{d_{\text{char}}} \right)^i \to \mathbb{R}^{d_{\text{tok}}} \tag{12}$$

$$s_{\text{attr}} : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{d_{\text{tok}}} \right)^i \to \mathbb{R}^{d_{\text{attr}}} \tag{13}$$

$$s_{\text{rec}} : \bigcup_{i=0}^{\infty} \left(\mathbb{R}^{d_{\text{attr}}} \right)^i \to \mathbb{R}^{d_{\text{rec}}} \tag{14}$$

The final hierarchical record embedding (HRE) model that we use to embed a record $r \in \mathcal{V}$ into the vector space $\mathbb{R}^{d_{\text{rec}}}$ is defined as:

$$h_{\text{rec}} = \text{comp}(s_{\text{tok}}, s_{\text{attr}}, s_{\text{rec}}) \circ \text{map}^3(e_{\text{char}}) \,. \tag{15}$$

2.2 Global Entity Matcher

We first introduce some definitions and notations, following the analysis in [18]:

Definition 5. for two vectors x_1 and x_2 in $\mathbb{R}^p$, we define the pairwise interaction features of x_1 and x_2 as the vector of $\mathbb{R}^{3p}$

$$\text{pwf}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \text{concat}\left(|x_1 - x_2|, \frac{1}{2}(x_1 + x_2), x_1 * x_2 \right) \tag{16}$$

where all the operations $+, -, *$ are applied element-wise.

Let us consider two records $(r_1, r_2) \in \mathcal{V}^2$ and their images (I_1, I_2) in the catalog. To process the images, we use pre-trained embeddings from a Resnet-50 backbone model trained with adaptive triplet loss on the images of the products in the catalog, following a

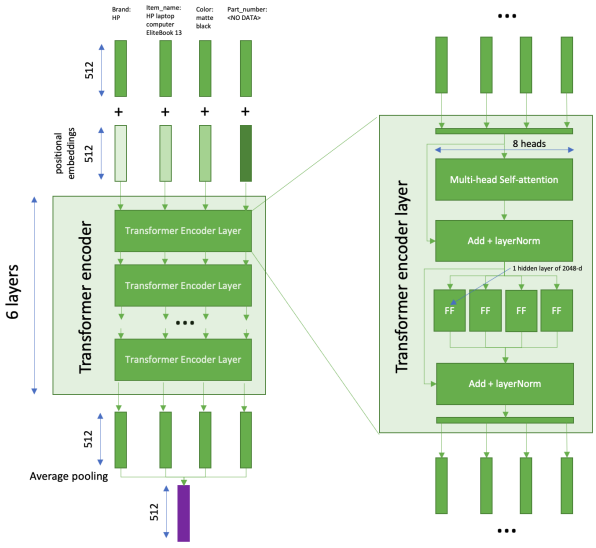

Figure 5: a transformer s_{rec}.

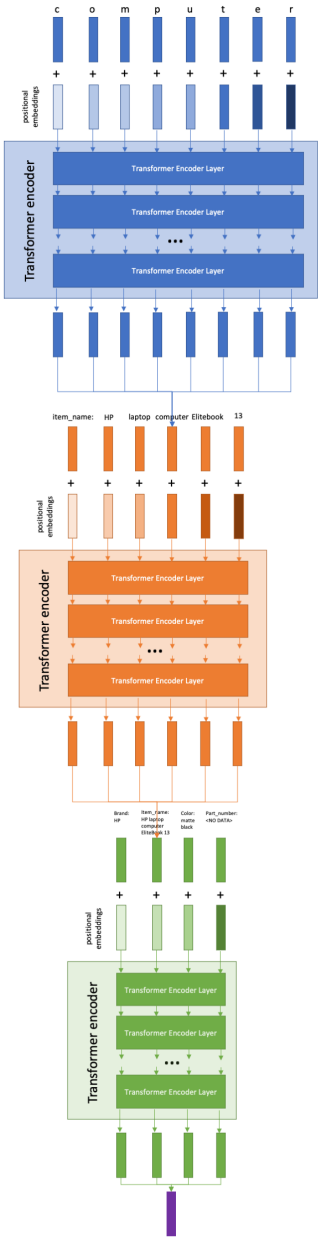

Figure 6: HRE with three-level hierarchical transformer.

similar approach to [22]. We denote the pretrained images embedding model e_{im}.

Remark 1. The images are optional and *not* required for matching. A lot of candidate product pairs for matching do not have an image at all (no image data), in either or both entities in the pair, and a lot of them have the exact same image in the pair despite being different, non-matching, entities (e.g. a shoe of size 10 will have the exact same image as the same shoe of size 11, but those are non-duplicate products). In the case in which an entity does not

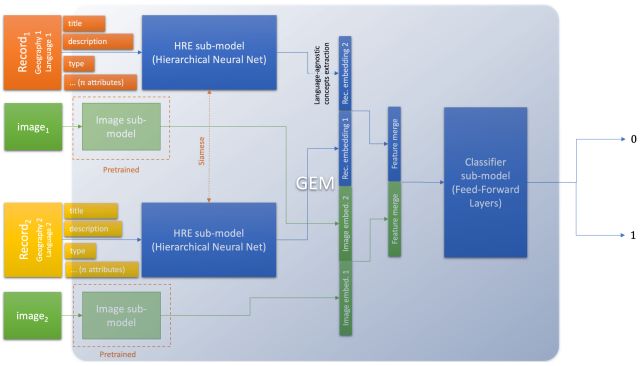

Figure 7: GEM model, fusing HRE embedding with image embedding then matching.

have any image data, we perform missing value imputation by imputing a neutral "all-gray" image to that entity, which then goes through the same GEM model as an entity with an image. All 4 scenarios are possible for ground-truth positive pairs: 1) with one of the entities missing image, 2) with both entities missing images, 3) with both entities having the same image, and 4) with both entities having distinct images. Similarly, all 4 scenarios are possible for ground-truth negative pairs.

The full features vector obtained from the records and their images is

$$f_{\text{rec,im}}\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \text{concat}\left(\text{pwf}\begin{pmatrix} h_{\text{rec}}(r_1) \\ h_{\text{rec}}(r_2) \end{pmatrix}, \text{pwf}\begin{pmatrix} e_{\text{im}}(I_1) \\ e_{\text{im}}(I_2) \end{pmatrix}\right). \quad (17)$$

This vector is fed to an MLP classifier with a final sigmoid activation to produce the probability of entity matching

$$P(r_1 \sim r_2) = \text{GEM}\begin{pmatrix} r_1 \\ r_2 \end{pmatrix} = \sigma\left(\text{mlp}\left(f_{\text{rec,im}}\begin{pmatrix} r_1 \\ r_2 \end{pmatrix}\right)\right) \quad (18)$$

The GEM model is trained against binary cross-entropy loss on a synthetic training dataset $\{(r_{i1}, r_{i2}, y_i)\}_i$ where y_i is the label (0 for no-match and 1 for match). The training optimizes for all the parameters Θ in e_{char}, s_{tok}, s_{attr}, s_{rec}, and mlp (the pretrained e_{im} is frozen)

$$\begin{aligned} \text{argmin}_\Theta - \sum_i y_i \log\left(\text{GEM}_\Theta\begin{pmatrix} r_{i1} \\ r_{i2} \end{pmatrix}\right) \\ + (1 - y_i)\log\left(1 - \text{GEM}_\Theta\begin{pmatrix} r_{i1} \\ r_{i2} \end{pmatrix}\right) \end{aligned} \quad (19)$$

3 TRAINING DATA SYNTHESIS

Let us suppose we have a dataset of labeled same-store EM pairs. We would like to synthesize from this dataset a new dataset composed of cross-language EM pairs suitable for GEM model training. We do this by exploiting the multilingual parallel data that occurs in some local catalogs records.

For example, multiple sellers in the US list their products in parallel in English (en) and Spanish (es), allowing us to extract cross-language en↔es GEM pairs from the US pairs. In Canada,

Table 1: Synthetic training dataset count.

Language pair	Synthesized from	Count
en↔es	US	8,969,102
en↔de	DE and US	4,996,649
de↔es	US	2,816,000
en↔fr	CA	429,254

multiple sellers list their products in parallel in English and in French, and in Germany multiple sellers list their products in in parallel in German and English.

For a given record r in a local catalog $\mathcal{C}$, we denote $\text{lang}(r)$ the set of languages in which the record r is listed. for a given language $l \in \text{lang}(r)$, we denote $r_{|l}$ the data associated with r in language l. For example, for a record r in the US store listed in both English and Spanish, $\text{lang}(r) = \{\text{en}, \text{es}\}$ and $r_{|\text{en}}$, $r_{|\text{es}}$ denote respectively the English data and the Spanish data in r.

Let now $(r_1, r_2, y) \in \mathcal{V}^2 \times \{0, 1\}$ be a labeled EM pair from one of the historical same-store local training datasets. We can synthesize from this pair a set of cross-language labeled pairs as follows

$$\text{xL}(r_1, r_2, y) = \big\{(r_{1|l_1}, r_{2|l_2}, y) \mid (l_1, l_2) \in \text{lang}(r_1) \times \text{lang}(r_2) \text{ and } l_1 \neq l_2\big\}. \quad (20)$$

Restricting the previous analysis to a specific language pair of interest (l_a, l_b), the set becomes

$$\text{xL}_{|(l_a,l_b)}(r_1, r_2, y) = \big\{(r_{1|l_1}, r_{2|l_2}, y) \mid (l_1, l_2) \in \big[\text{lang}(r_1) \cap \{l_a, l_b\}\big] \times \big[\text{lang}(r_2) \cap \{l_a, l_b\}\big] \text{ and } l_1 \neq l_2\big\} \quad (21)$$

Let now $\mathcal{M} = \{\text{US}, \text{UK}, \text{Canada}, \text{France}, \ldots\}$ denote the set of all local stores and $\mathcal{D}_m$ denote the local EM training data in store $m \in \mathcal{M}$. The synthesized training dataset for a given language pair (l_a, l_b) is

$$\mathcal{D}_{\text{GEM},(l_a,l_b)} = \bigcup_{m\in\mathcal{M}} \bigcup_{(r_1,r_2,y)\in\mathcal{D}_m} \text{xL}_{|(l_a,l_b)}(r_1, r_2, y) \quad (22)$$

and the synthesized global training dataset

$$\mathcal{D}_{\text{GEM}} = \bigcup_{m\in\mathcal{M}} \bigcup_{(r_1,r_2,y)\in\mathcal{D}_m} \text{xL}(r_1, r_2, y) \quad (23)$$

4 EXPERIMENTS AND RESULTS

We experimented with three GEM models where

$$(s_{\text{tok}}, s_{\text{attr}}, s_{\text{rec}}) \in \{(\text{BiLSTM}_{\text{tok}}, \text{BiLSTM}_{\text{attr}}, \text{BiLSTM}_{\text{rec}}), (\text{seqCNN}_{\text{tok}}, \text{seqCNN}_{\text{attr}}, \text{seqCNN}_{\text{rec}}), (\text{seqCNN}_{\text{tok}}, \text{seqCNN}_{\text{attr}}, \text{Transformer}_{\text{rec}})\} \quad (24)$$

All the models were instantiated using PyTorch.

The training and test datasets for the GEM model were obtained from the data of a global shopping website, the schema of the listings includes attributes such as *title, description, type, etc.*

The local EM training dataset $\bigcup_{m\in\mathcal{M}} \mathcal{D}_m$ is composed of a sample of around 10 million labeled pairs, with around 30% positives. From that dataset we synthesized a GEM training dataset of around 20 million cross-language pairs, containing en↔es, en↔de, es↔de, en↔fr pairs, see Table 1. Random dev and test set were sampled using stratified-sampling per language pair and per label (50% positives in each language-pair) in a classical train-dev-test split. The dev and test sets contain 10,000 pairs each, 1,250 from each class in each language pair, the rest is in the train set. To test against actual wild data and not synthetic data, we additionally manually-labeled (by expert auditors) a wild-test set with UK↔Germany cross-store sampled pairs where only the English data in UK records and the German data in Germany records were used for auditing and for model evaluation. This wild-test set contains around 5,000 pairs with around 10% positives, which we upsampled to have 50%-50% positive-negative distribution and get numbers that are comparable to the test and dev sets. We trained each model once for about 1 epoch using Adam optimizer with learning rate 1e-5 for the BiLSTM model and 1e-3 for all other models, on 8 NVIDIA V100 Tensor Core GPUs with 16GB GPU memory in each. All reported results are Area under the Precision-Recall Curve (PR-AUC), which baselines at 50% for all the tests given the balanced distribution. To test the zero-shot generalization capability of the model to pairs of languages unseen in training, we additionally trained models while purposely holding out one of the language pairs from the training mix. The zero-shot tests yielded the same performance as the test sets of the language pairs from the training mix. See Table 3 for the results.

Since there was no existing GEM baseline to compare against, we compared the GEM model trained in a single local store (US), with a state-of-the-art English-only EM model. Table 2 shows the results for this baseline comparison, serving also as a comparison between the three different instantiations of the HRE model.

5 DISCUSSION

In this section, we discuss the architectural choices and the contribution of the paper, which is 3-fold: **i**. The HRE model; to embed arbitrarilyhigh-order tensors modeling the entities as a cascade of nested sequences (in our case order-4 tensors) into a vector space (order-1 tensors), progressively reducing each dimension of the tensor with an independent sequence embedding model. **ii**. The GEM model; which combines a Siamese HRE model with a pre-trained image-feature extractor network to compute the probability of matching of the two entities. **iii**. The training data synthesis approach; to leverage existing same-store labeled data only. The model trained with the synthetic data proved transferable to the real data even in zero-shot inference setting.

Two important characteristics allow the GEM model to solve the translation-free cross-lingual Global Entity Matching problem: **1**. having a textual record embedding model powerful enough to capture the exact identity at the *single-entity* level with fine-enough resolution, removing the need for attribute-pair matching features at the *entity-pair* level, commonly required in existing EM models. **2**. Having a shared vocabulary between the languages, small enough to be computationally tractable across all languages yet rich enough to capture important rare named entities and codes that contribute to the exact identity of the product.

The HRE model satisfies characteristic **1** above. It is a powerful deep-learning feature extractor that encodes the *exact identity* of

Table 2: Comparison against English-only baseline SOTA model.

	all	hardlines	softlines	consumables	media
EM baseline, no images	91.5%	91.5%	90.6%	89.0%	91.5%
GEM BiLSTM, no images	92.3%	90.3%	86.5%	88.3%	94.1%
GEM SeqCNN, no images	92.2%	92.0%	88.6%	87.7%	91.1%
GEM Transformer, no images	92.6%	91.0%	88.2%	89.3%	92.6%
GEM SeqCNN, with images	93.1%	94.1%	93.1%	94.7%	94.0%

Table 3: GEM models trained on various combinations of languages, zero-shot tests are highlighted in bold red

	training mix	test en↔de	test en↔es	test de↔es	test en↔fr	wild-test en↔de
GEM SeqCNN, with images	en↔de	86.4%	-	-	-	94.0%
GEM SeqCNN, with images	en↔de, en↔es	85.2%	86.7%	**85.3%**	-	-
GEM SeqCNN, with images	es↔de, en↔es	**84.2%**	86.6%	88.4%	-	**94.2%**
GEM SeqCNN, with images	es↔de, en↔es, en↔fr	**84.6%**	85.8%	87.5%	86.7%	**93.2%**
GEM SeqCNN, with images	en↔de, en↔es, es↔de, en↔fr	85.0%	87.4%	86.2%	88.6%	94.6%

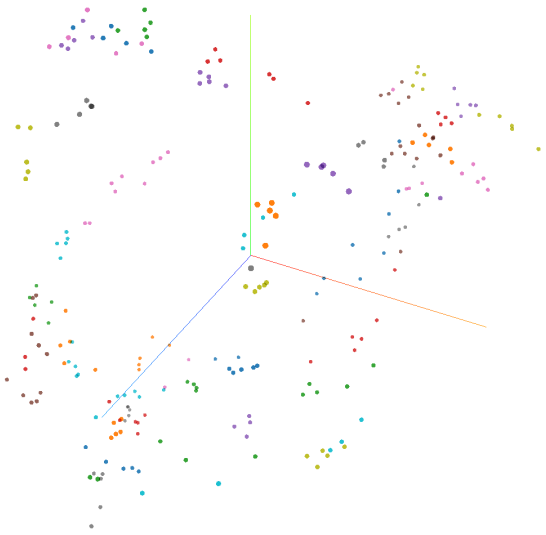

Figure 8: UMAP visualization of the h_{rec} embeddings for quintuplets of same products listed in different local stores (UK, Germany, France, Spain, Italy) (colored by entity, with some duplicate colors). The representations of the same entity across different languages are generally aligned in the embedding space.

an entity in a single vector. The exact identity is a fine-resolution concept that can be altered with minute detail variations such as color or size variation, or even by a single character in a model-number. Yet, although being sensitive to these details, the HRE model had to be robust against completeness, correctness, and consistency issues in the seller-provided data (empty fields, gibberish data, placeholder data, data input in wrong fields, etc.). Thanks to the third level in its hierarchy of sequence embeddings, namely the s_{rec} level, the HRE model captures the semantic interactions between the different attributes and learns to overcome the completeness and consistency issues between them. Such characteristic is absent from existing product-embedding deep-learning models used in tasks such as product recommendation or product category classification, which flatten at some point the attributes by concatenating their embeddings, or by running an NLP embedding model directly on the concatenated attributes.

Additionally, the character-level nature of the HRE model contributes to both characteristics **1** and **2**. The union of character vocabularies is small as characters are mostly shared among the different languages, allowing to train the GEM model transparently on pairs of any respective source languages without needing to know what the languages are. Therefore, the model is robust against instances in which the seller lists the record in a language different from the declared language tag. The character-level model also allows to handle the open-vocabulary nature of the catalog. In the model-number example, the GEM model is able to understand the differences between the two model-numbers and learns to distinguish between gibberish data and correct data. A word-level model would fail in the same task, unless augmented with additional hand-crafted features and with careful hashing mechanisms of the OOV tokens. The open-vocabulary nature of the catalogs also makes the use of multi-lingual aligned word embeddings such as MUSE [8] not directly adapted to the problem.

6 CONCLUSION

Global entity matching had never been addressed in the literature before. We proposed a translation-free, end-to-end approach to tackle the GEM problem using only same-store training datasets. The model performed on par with English-only state-of-the-art EM models on English only datasets and yielded comparable performances on GEM tasks. At the core of the model, we introduced a language-agnostic character-level record embedding model using a multi-level hierarchical record embedding model. A single model

handled multiple language-pairs and generalized well to zero-shot inference tasks on language pairs unseen in the training dataset. Future work will address the problem of non-alphabetical languages (Japanese, Chinese).

REFERENCES

[1] Arijit Biswas, Mukul Bhutani, and Subhajit Sanyal. 2017. Mrnet-product2vec: A multi-task recurrent neural network for product embeddings. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 153–165.

[2] William Cohen, Pradeep Ravikumar, and Stephen Fienberg. 2003. A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, Vol. 3. 73–78.

[3] Vincenzo Di Cicco, Donatella Firmani, Nick Koudas, Paolo Merialdo, and Divesh Srivastava. 2019. Interpreting deep learning models for entity resolution: an experience report using LIME. In *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. 1–4.

[4] Halbert L Dunn. 1946. Record linkage. *American Journal of Public Health and the Nations Health* 36, 12 (1946), 1412–1416.

[5] Ahmed K Elmagarmid, Panagiotis G Ipeirotis, and Vassilios S Verykios. 2006. Duplicate record detection: A survey. *IEEE Transactions on knowledge and data engineering* 19, 1 (2006), 1–16.

[6] Ivan P Fellegi and Alan B Sunter. 1969. A theory for record linkage. *J. Amer. Statist. Assoc.* 64, 328 (1969), 1183–1210.

[7] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073* (2016).

[8] Edouard Grave, Armand Joulin, and Quentin Berthet. 2018. Unsupervised alignment of embeddings with wasserstein procrustes. *arXiv preprint arXiv:1805.11222* (2018).

[9] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics* 5 (2017), 339–351.

[10] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[11] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[12] Sebastian Klenk, Dennis Thom, and Gunther Heidemann. 2009. The normalized compression distance as a distance measure in entity identification. In *Industrial Conference on Data Mining*. Springer, 325–337.

[13] Hanna Köpcke, Andreas Thor, Stefan Thomas, and Erhard Rahm. 2012. Tailoring entity resolution for matching product offers. In *Proceedings of the 15th International Conference on Extending Database Technology*. 545–550.

[14] Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. *arXiv preprint arXiv:1508.02096* (2015).

[15] Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. *arXiv preprint arXiv:1804.08198* (2018).

[16] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*. 19–34.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 8024–8035.

[18] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[19] Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. *arXiv preprint arXiv:1704.04154* (2017).

[20] Adelene YL Sim and Andrew Borthwick. 2018. Record2Vec: unsupervised representation learning for structured records. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1236–1241.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[22] Xiaonan Zhao, Huan Qi, Rui Luo, and Larry Davis. 2019. A weakly supervised adaptive triplet loss for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 0–0.