LinkTransformer: A Unified Package for Record Linkage with Transformer Language Models

Abhishek Arora and Melissa Dell*
Harvard University, Cambridge, MA, USA
*Corresponding author: melissadell@fas.harvard.edu.

Abstract

Linking information across sources is fundamental to a variety of analyses in social science, business, and government. While large language models (LLMs) offer enormous promise for improving record linkage in noisy datasets, in many domains approximate string matching packages in popular softwares such as R and Stata remain predominant. These packages have clean, simple interfaces and can be easily extended to a diversity of languages. Our open-source package `LinkTransformer` aims to extend the familiarity and ease-of-use of popular string matching methods to deep learning. It is a general purpose package for record linkage with transformer LLMs that treats record linkage as a text retrieval problem. At its core is an off-the-shelf toolkit for applying transformer models to record linkage with four lines of code. `LinkTransformer` contains a rich repository of pre-trained transformer semantic similarity models for multiple languages and supports easy integration of any transformer language model from Hugging Face or OpenAI. It supports standard functionality such as blocking and linking on multiple noisy fields. `LinkTransformer` APIs also perform other common text data processing tasks, *e.g.*, aggregation, noisy de-duplication, and translation-free cross-lingual linkage. Importantly, `LinkTransformer` also contains comprehensive tools for efficient model tuning, to facilitate different levels of customization when off-the-shelf models do not provide the required accuracy. Finally, to promote reusability, reproducibility, and extensibility, `LinkTransformer` makes it easy for users to contribute their custom-trained models to its model hub. By combining transformer language models with intuitive APIs that will be familiar to many users of popular string matching packages, `LinkTransformer` aims to democratize the benefits of LLMs among those who may be less familiar with deep learning frameworks.

1 Introduction

Linking information across sources is fundamental to a variety of analyses in social science, business, and government. A recent literature, focused on matching across e-commerce datasets, shows the promise of transformer large language models (LLMs) for improving record linkage (alternatively termed entity resolution or approximate dictionary matching). Yet these methods have not yet made widespread inroads, with rule-based methods continuing to overwhelmingly predominate in social science and government applications (*e.g.*, see reviews by Binette and Steorts (2022); Abramitzky et al. (2021)). In particular, users commonly employ string-based matching tools available in statistical software packages such as R or Stata.

We suspect that record linkage with large language models has not made further inroads at least in part due to the lack of packages that match the ease of the popular string matching packages, which are intuitive, extensible, and easy-to-use. They require little coding expertise and can easily be applied across different languages and settings. In contrast, existing tools for large language model matching require considerable technical expertise to implement. This makes sense in the context for which these models were developed - classifying and linking products for e-commerce firms, which employ data scientists - but it is a significant impediment for broader use.

To bridge the gap between the ease-of-use of widely employed string matching packages and the power of modern LLMs, we developed `LinkTransformer`, a general purpose, user friendly package for record linkage with transformer LLMs. `LinkTransformer` treats record linkage as a text retrieval problem (See Figure 1). The API can be thought of as a drop-in replacement to popular dataframe manipulation frameworks like pandas or tools like R and Stata, catering to those

who lack extensive exposure to coding.

To achieve its objective of democratizing access to the benefits of deep learning amongst those who may lack familiarity with deep learning frameworks, `LinkTransformer` integrates the following features:

1. An off-the-shelf toolkit for applying transformer models to record linkage and de-duplication with 4 lines of code
2. A rich repository of pre-trained semantic similarity models, supporting multiple languages, that underlies off-the-shelf usage
3. Easy integration of any language transformer model on Hugging Face or OpenAI
4. APIs to support related data processing tasks, *e.g.*, aggregation, de-duplication, and translation-free cross-lingual linkage
5. Comprehensive tools for efficient model tuning to facilitate different levels of customization
6. Easy sharing of models, to promote reusability, reproducibility, and extensibility

The `LinkTransformer` model zoo currently contains English, Chinese, French, German, Japanese, Spanish, and multilingual pre-trained models. We initialize with semantic similarity models, *e.g.,* Reimers and Gurevych (2019), which have desirable properties relative to using off-the-shelf embeddings from models like RoBERTa (see Section 2). We further tuned these models on a variety of linked datasets.

While transfer learning can facilitate strong off-the-shelf performance in many scenarios, supporting customization is important. Record linkage applications are extremely diverse in their languages, time periods, and domains, which vary significantly in how out-of-domain they are from the web corpora that underlying LLMs are trained on. Considerable heterogeneity - combined with settings that demand extremely high accuracy - create many scenarios where custom training is useful.

We show that `LinkTransformer` performs well on challenging record linkage tasks. It is equally applicable to record linkage tasks with a single field - *e.g.,* linking 1940s Mexican tariff product classes across time - and applications that require concatenating an array of noisily measured fields - *e.g.,* linking 1950s Japanese firms across different large-scale, noisy databases using the firm name, location, products, shareholders, and banks. This type of linkage problem would be highly convoluted with traditional string matching methods, as there are many noisily measured fields of relevance (*e.g.,* products can be described in different ways, different subsets of managers and shareholders are listed, etc). Using `LinkTransformer` to automatically concatenate the information and feed it to a LLM handles these challenges with ease. A demo is available at `https://www.youtube.com/watch?v=Sn47nmCvV9M`. More resources are available on our package website `https://linktransformer.github.io/`.

`LinkTransformer` has a GNU General Public License. It is being actively maintained, and in the next release we will add support for vision-free and multimodal linkage models, including support to import and customize any timm model (Wightman, 2019). When OCR errors are rampant, vision-only or aligned vision-language transformer models can improve record linkage, relative to string matching or language-only transformer linking *e.g.*, (Yang et al., 2023; Arora et al., 2023).

The rest of the paper is organized as follows. Section 2 provides an overview of related work. The core `LinkTransformer` library, Model Zoo, customized model training, and model sharing are described in Section 3. Section 4 discusses various use cases. Section 5 discusses limitations.

2 Relation to the Existing Literature

There is a large literature on record linkage spanning social science, statistics, and computer science. Record linkage serves as a prerequisite for many empirical analyses, which often require combining text data from multiple noisy sources. In social science, statistics, and government applications, large language models have made few inroads. A 2022 review of the record linkage literature in *Science Advances* (Binette and Steorts, 2022), entitled "(Almost) All of Entity Resolution", concludes that deep neural models are unlikely to be applicable to record linkage using structured data, arguing that training datasets are small and there is unlikely to be much gained from large language models since text fields are often short.

While there are undoubtedly linking tasks for which LLMs will not be of much use (discussed further in Section 5 on limitations), an extensive

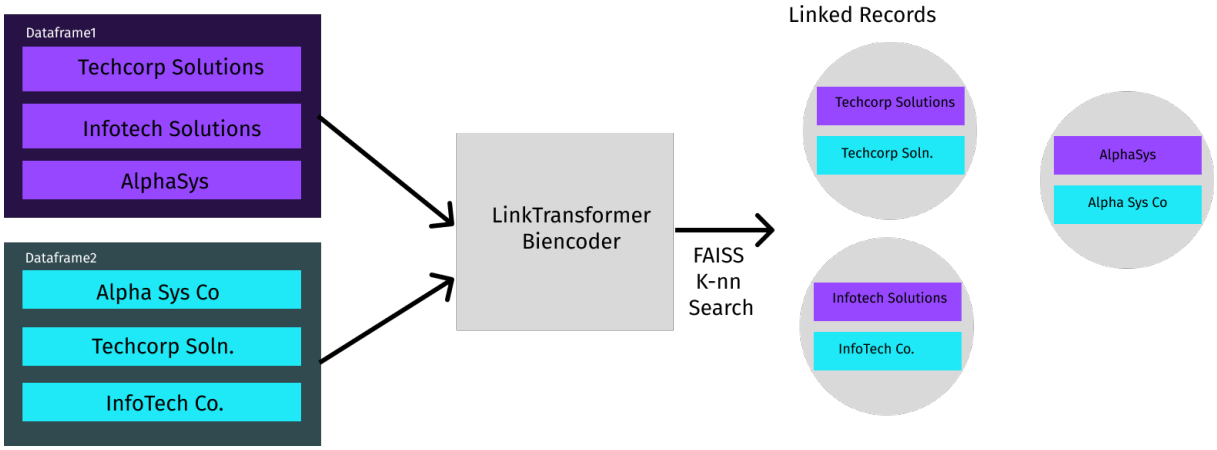

Figure 1: **Visualization.** This figure shows the `LinkTransformer` architecture.

literature on e-commerce applications underscores their utility for linking structured datasets. Benchmarks in this literature (e.g., Köpcke et al. (2010); Das et al. (2015); Primpeli et al. (2019)) focus on applications such as matching electronics and software products between Amazon-Google and Walmart-Amazon listings, matching iTunes and Amazon music listings, and matching restaurants between Fodors and Zagats. The focus is on applications in English. Recent studies have used masked language models (e.g., BERT, DistilBERT, RoBERTa) (Li et al., 2020; Joshi et al., 2021; Brunner and Stockinger, 2020), GPT (Peeters and Bizer, 2023; Tang et al., 2022), or both, significantly outperforming static word embedding and other older linkage methods. Zhou et al. (2022), like `LinkTransformer`, uses Sentence BERT (Reimers and Gurevych, 2019).

The main package in this space to our knowledge, Ditto (Li et al., 2020), implements the methods later published in Li et al. (2023). It requires significant programming expertise to deploy, appropriate for a litearture focused on e-commerce - where data scientists predominate - but a hindrance for broader use.

Most of the literature examining record linkage with LLMs poses record linkage as a classification task, which is appropriate for the e-commerce benchmarks. However, this significantly limits extensibility, as in many social science and government applications the number of entities to be linked numbers in the millions, making it computationally infeasible to compute a softmax over all possible classes (entities).

`LinkTransformer` frames record linkage as a knn-retrieval task, in which the nearest neighbor for each entity in a query embedding dataset is retrieved from a key embedding dataset, using cosine similarity implemented with an FAISS backend (Johnson et al., 2019). `LinkTransformer` includes functionality to tune a no-match threshold - since not all entities in the query need to have a match in the key - and allows for retrieving multiple neighbors, to accommodate many-to-many matches between the query and the key. The `LinkTransformer` architecture was inspired by a variety of bi-encoder applications for unstructured texts, *e.g.*, passage retrieval (Karpukhin et al., 2020), entity disambiguation (Wu et al., 2019), and entity co-reference resolution (Hsu and Horwood, 2022).

`LinkTransformer` departs from much of the literature (with the exception of Zhou et al. (2022)) in utilizing LLMs trained for semantic similarity for its pre-trained models.[1] A large literature shows that off-the-shelf LLMs such as BERT have anisotropic geometries (Ethayarajh, 2019): representations of low frequency words are pushed outwards on the hypersphere, the sparsity of low frequency words violates convexity, and the distance between embeddings is correlated with lexical similarity. This leads to poor performance when individual term representations from the transformer model are pooled to create a representation for longer texts - as is often necessary for record linkage - since pooling assumes convexity, and leads to poor alignment between semantically similar texts. Contrastive training for semantic similarity reduces anisotropy, improving alignment between semantically similar pairs and improv-

[1] Compare to Zhou et al. (2022), `LinkTransformer` uses a different loss function, supervised contrastive loss (Khosla et al., 2020). It is well-suited to record linkage as there are multiple positive pairs in the datasets we use for pre-training.

ing sentence embeddings (Wang and Isola, 2020; Reimers and Gurevych, 2019). `LinkTransformer` builds closely upon Sentence BERT (Reimers and Gurevych, 2019), whose excellent semantic similarity library inspired many of the features in `LinkTransformer`.

The knn retrieval structure of `LinkTransformer` also supports noisy de-duplication, a closely related task that finds noisily duplicated observations within a dataset. `LinkTransformer` follows the methods developed in Silcock et al. (2023), who show that de-duplication using a contrastively trained bi-encoder significantly outperforms n-gram and locally sensitive hashing methods and is highly scalable.

3 The `LinkTransformer` Library

3.1 Off-the-shelf Toolkit

At the core of `LinkTransformer` is an off-the-shelf toolkit that streamlines record linkage with transformer language models. The record linkage models enable using pre-trained or self-trained transformer models with just 4 lines of code. Any Hugging Face or OpenAI model can be used by configuring the `model` and `openai_key` arguments. This future-proofs the package, allowing it to take advantage of the open-source revolution that Hugging Face has pioneered. Here is an example of the core **merge** functionality, based on embeddings sourced from an external language model.

```
import linktransformer as lt
#Load data frame
df1 = pd.read_csv("df1.csv")
df1 = pd.read_csv("df2.csv")
df_matched = lt.merge(df2, df1,
    merge_type='1:m', on=["Varname"],
    model="sentence-transformers/all-
    MiniLM-L6-v2", openai_key=None)
```

`LinkTransformer` provides a wealth of pre-trained model weights, covering different languages and domains. It currently supports six languages (English, Chinese, French, German, Japanese, and Spanish) with both multilingual and monolingual models. Training datasets include:

1. A novel dataset of firm aliases that we compiled from Wikidata for 6 languages

2. United Nations economic classification schedules (International Standard Industrial Classification, Standard International Trade Classification, and Central Product Classification), which homogenize industry and product classifications across varying national systems. We include models trained on these for 3 of the 6 official languages of the UN.

3. A variety of e-commerce linking datasets (see Supplemental Materials)

We name these models with a semantic syntax: `{org_name}/lt-{data}-{task}-{lang}`. Each model has a detailed model card, with the appropriate tags for quick model discovery. Additionally, for the tasks we trained our models on, we provide a high-level interface to download the right model by task through a wrapper that retrieves the best model for a task chosen by the user.

`LinkTransformer` makes no compromise in scalability. All functions are vectorized wherever possible and the vector similarity search underlying knn retrieval is accelerated by an FAISS (Johnson et al., 2019) backend that can easily be extended to perform retrieval on GPUs on massive datasets. We also allow "blocking" - running knn-search only within "blocks" that can be defined by the `blocking_vars` argument.

Record linkage frequently requires matching databases on multiple noisily measured keys. `LinkTransformer` allows a list of as many variables as needed in the "on" argument. The merge keys specified by the on variable are serialized by concatenating them with a $< SEP >$ token, which is based on the underlying tokenizer of the selected base language model. This ensures that the serialization takes advantage of a token already introduced in training and the process is agnostic to the choice of the model.

Since we have designed the API around dataframes - due to their familiarity amongst users of R, Stata or Excel - all import/export formats supported.

The accessible `LinkTransformer` API supports a plethora of other features that are an essential part of routine data analysis pipelines. These include:

Aggregation: Data processing often requires the aggregation of fine descriptions into coarser categories, that are consistent across datasets/time or facilitate interpretation. This problem can be thought of as a merge between finer categories and coarser ones, where `LinkTransformer` classifies the finer categories by means of finding their nearest coarser neighbor(s). We provide a high-level API for this task, `lt.aggregate_rows`, with a similar syntax to the main record linkage API.

Deduplication: Text datasets can contain noisy duplicates. Popular libraries like dedupe (Gregg and Eder, 2022) only support deduplication using metrics that most closely resemble edit distance. `LinkTransformer` allows for semantic deduplication with a single, intuitive function call.

```
df=pd.read_csv("df1.csv"))
df_dedup=lt.dedup_rows(df,on="
    CompanyName",model="sentence-
    transformers/all-MiniLM-L6-v2",
    cluster_params= {'threshold': 0.7})
```

`LinkTransformer` de-duplication clusters embeddings under the hood, with embeddings in the same cluster classified as duplicates. `LinkTransformer` supports several clustering methods like SLINK, DBSCAN, HDBSCAN, and agglomerative clustering.

Cross-lingual linkage: Analyses spanning multiple countries often require cross-lingual linkage. This task typically requires machine translation followed by a merge. Edit distance metrics tend to do particularly poorly in this scenario, necessitating costly hand linking. `LinkTransformer` users can bypass translation by using multilingual transformer models.

We provide helpful notebooks and tutorials on the `LinkTransformer` website https://linktransformer.github.io/ to outline the use of these functionalities and also some toy datasets. We also have a tutorial to help those who are less familiar with language models to select ones that best fit their use case. More detailed information and additional features that we cannot highlight in the interest of brevity can be found in the online `LinkTransformer` documentation that is available on our public repo.[2]

3.2 Customized Model Training

Record linkage tasks are highly diverse. Hence, `LinkTransformer` also supports easy model training. Custom training can be initialized using the weights of any Hugging Face transformer model.

Training data are expected in a *pandas* data frame, removing entry barriers for the typical social science user who is familiar with other programming languages and statistical packages. A data frame can include only positive labeled examples (linked observations) as inputs, in which case the model is evaluated using an information retrieval evaluator that measures top-1 retrieval accuracy. Alternately, it can take a list of both positive and negative pairs, in which case the model is evaluated using a binary classification objective.

Only the most important arguments are exposed and the rest have reasonable defaults which can be tweaked by more advanced users. Additionally, `LinkTransformer` supports logging of a training run on Weights and Biases (Biewald, 2020).

```
best_model_path=lt.train_model(
        model_path="hf-path-to-base-
    model",
        data="df1.csv"),
        left_col_names=["left_var"],
        right_col_names=['right_var'],
        left_id_name=['left_id'],
        right_id_name=['right_id'],
        label_col_name=None,
        log_wandb=False,
        training_args={"num_epochs": 1}
    )
```

Default training expects positive pairs. A simple argument that specifies `label_col_name` switches the dataset format and model evaluation to adapt to positive and negative labels. To make this extensible to most record linkage use-cases, the model can also be trained on a dataset of cluster ids and texts by simply specifying `clus_id_col_name` and `clus_text_col_names`.

`LinkTransformer` is sufficiently sample efficient that most models in the model zoo were trained with a student Google Colab account, an integral feature since the vast majority of potential users have constrained compute budgets.

3.3 User Contributions

`LinkTransformer` aims to promote the reusability and reproducibility of record linkage pipelines. End-users can upload their self-trained models to the `LinkTransformer` Hugging Face hub with a simple `model.save_to_hub` command. Whenever a model is saved, a model card is automatically generated that follows best practices outlined in Hugging Face's Model Card Guidebook. This process adds a pipeline-tag, supported language(s) (given the base model), and other tags to the model card's header that facilitate model discovery by other Hugging Face users. Moreover, the automatically generated card contains instructions on how to use the model for record linkage and model-specific architecture and training details.

4 Applications

The LLMs in the `LinkTransformer` model zoo outperform non-neural methods by a wide margin.

[2] https://github.com/dell-research-harvard/linktransformer

We examine applications from both modern web data and historical datasets that we digitized from hard copy historical firm and government records. These tasks are representative of applications in quantitative social science, industry, and government. The results are reported in the supplementary materials.

First, we evaluate in-domain performance on semantic similarity datasets used to pre-train `LinkTransformer` models, with the supplementary materials comparing the accuracy of Levenshtein edit distance matching (Levenshtein et al., 1966), semantic similarity models off-the-shelf, and `LinkTransformer` tuned models. The `LinkTransformer` and OpenAI models tend to outperform edit distance metrics by a wide margin in matching Wikidata firm aliases, with the top-1 retrieval accuracy tending to be 20-30 points higher for `LinkTransformer` linkage than for edit distance based matching. Cases that `LinkTransformer` gets wrong are often impossible even for a skilled human to resolve from the firm names alone, *e.g.,* in cases where a firm is referred to by two completely disparate acronyms. Off-the-shelf semantic similarity models also tend to outperform string-matching methods, albeit not by as much of a margin as tuned models.

Second, we examine historical applications. The first setting is a concordance between products in two 1940s Mexican tariffs schedules, published by the Mexican government and digitized from the hard copy documents (Secretaria de Economía de Mexico, 1948). Tariffs were applied at an extremely disaggregated product level and each of the many thousands of products in the tariff schedule is identified only by a text description, which can change each time the tariff schedule is updated. Around 2,000 products map to different descriptions across the schedules. While there are considerable debates on the role that trade policies have played in long-run development, empirical evidence is limited in part due to the considerable challenges in homogenizing extremely detailed historical tariff schedules across time, as crosswalks do not exist for most tariff schedule changes.

We link the tariff schedules using an off-the-shelf semantic similarity model, as well as a model tuned on the in-domain historical data and the recommended Open AI embeddings (from the model *text-embedding-ada-002*). All transformer models widely outperform edit distance, with the fine-tuned `LinkTransformer` model (at no charge) and purchased GPT hidden states producing similar, near perfect results.

We also link firms across two different 1950s publications created by different Japanese credit bureaus (Jinji Koshinjo, 1954; Teikoku Koshinjo, 1957). One has around 7,000 firms and the other has around 70,000, including many small firms. Firm names can be written differently across publications and there are many duplicated or similar firm names. To make this task feasible, we concatenate information on the firm's name, prefecture, major products, shareholders, and banks. These variables contain OCR noise and the information included in each publication varies somewhat, *e.g.* in terms of which products are described, which shareholders are included, etc. This makes rule-based methods quite brittle. Again, neural models significantly outperform string matching methods, with OpenAI giving the best results. (Due to the cost of labeling, our training dataset may not be large enough to capture the benefits of tuning.)

Finally, we examine the various e-commerce and industry benchmarks that prevail in this literature. We used exactly the same training procedure for each benchmark, to avoid overfitting, which is often not the case in the literature. We have generally comparable performance, sometimes outperformed by other models (that could be integrated into `LinkTransformer` as well if on Hugging Face) and sometimes outperforming other models.

5 Limitations

`LinkTransformer` is built upon transformer language models, and hence will not be suitable for lower resource languages that lack pre-trained LLMs. `LinkTransformer` will also be less useful in contexts where little language understanding enters record linkage, *e.g.*, when linking records solely using individual names. For these contexts, the next release of `LinkTransformer` will integrate vision-only transformer models, which the authors have reported on extensively elsewhere (Arora et al., 2023; Yang et al., 2023).

Furthermore, in settings where OCR errors are considerable, too much information may have been destroyed to successfully link entities using garbled texts. In this case, a multimodal framework (*e.g.*, Arora et al. (2023)) that uses aligned language and vision models to incorporate the original image crops or a matching framework that incorpo-

rates character visual similarity (Yang et al., 2023) - as OCR errors tend to confuse visually similar characters - may be required. This again will be incorporated into `LinkTransformer`.

`LinkTransformer` relies on backends that many end users - accustomed primarily to working with statistical packages like Stata or R - will not be familiar with. We recommend that users new to LLMs deploy the package using a cloud service optimized for deep learning, such as Colab, to avoid the need to resolve dependencies. To reduce startup costs, we will provide detailed tutorials for `LinkTransformer` instillation, inference, and training on Colab.

Supplementary Materials

S-1 Training and other details

`LinkTransformer` models use AdamW as the optimizer with a linear schedule with a 100% warm-up with 2e-6 as the max learning rate. We use a batch size of 64 for models trained with Wikidata (companies) and UN data (products). For industry benchmarks, we used a batch size of 128. We trained the models for 150 epochs for industrial benchmarks and 100 epochs for UN/Wikidata/Historic applications. We used Supervised Contrastive loss (Khosla et al., 2020) as the training objective with default hyperparameters. The implementation for the loss was based on the implementation shared on the sentence-transformers repository (Reimers and Gurevych, 2019).

`LinkTransformer` uses IndexFlatIP from FAISS (Johnson et al., 2019) as the index of choice, allowing an exhaustive search to get k nearest neighbours. We use the inner-product as the metric. All embeddings from the encoders are L2-normalized such that the distances (inner-products) given by the FAISS indices are equivalent to cosine similarity.

Code to replicate the below tables and train the models are available on our repository, which also contains links to our training data.

S-2 Datasets and Results

Table S-1 lists the base sentence transformer models that we used to initialize the custom `LinkTransformer` models. Table S-2 describes the datasets used for training the `LinkTransformer` model zoo. They are drawn from multilingual UN product and industry concordances, Wikidata company aliases, a 1948 Mexican government concordance between tariff schedules (Secretaria de Economía de Mexico, 1948), and a hand-linked dataset between two 1950s Japanese firm-level datasets collected by credit bureaus, one containing around 7,000 firms and the other around 70,000 (Teikoku Koshinjo, 1957; Jinji Koshinjo, 1954). Table S-3 describes the train-val-test splits for each of these datasets.

Table S-4 shows results of fine-tuned models, where the models are evaluated on the test split of the dataset they were trained on. Table S-5 examines performance on two historical linkage tasks: linking historical Mexican tariff schedules and linking historical Japanese firms across large, noisy databases. Table S-6 reports results on standard industry and e-commerce benchmarks for record linkage.

These results - discussed in the main text - show that deep neural record linkage (whether with off-the-shelf semantic similarity models, OpenAI models, or custom `LinkTransformer` models) outperforms Levenshtein edit distance by a wide margin.

Language	Base Model
English	sentence-transformers/multi-qa-mpnet-base-dot-v1
Japanese	oshizo/sbert-jsnli-luke-japanese-base-lite
French	dangvantuan/sentence-camembert-large
Chinese	DMetaSoul/sbert-chinese-qmc-domain-v1
Spanish	hiiamsid/sentence_similarity_spanish_es
German	Sahajtomar/German-semantic
Multilingual	sentence-transformers/paraphrase-multilingual-mpnet-base-v2

Table S-1: We used the above sentence-transformers models for different langauges as base models to train `LinkTransformer` models. They were selected from the Hugging Face model hub and the names correspond to the repo names on the Hub.

Model	Training Data
lt-wikidata-comp-en	Wikidata English-language company names.
lt-wikidata-comp-fr	Wikidata French-language company names.
lt-wikidata-comp-de	Wikidata German-language company names.
lt-wikidata-comp-ja	Wikidata Japanese-language company names.
lt-wikidata-comp-zh	Wikidata Chinese-language company names.
lt-wikidata-comp-es	Wikidata Spanish-language company names.
lt-wikidata-comp-multi	Wikidata multilingual company names (en, fr, es, de, ja, zh).
lt-wikidata-comp-prod-ind-ja	Wikidata Japanese-language company names and industries.
lt-un-data-fine-fine-en	UN fine-level product data in English.
lt-un-data-fine-coarse-en	UN coarse-level product data in English.
lt-un-data-fine-industry-en	UN product data linked to industries in English.
lt-un-data-fine-fine-es	UN fine-level product data in Spanish.
lt-un-data-fine-coarse-es	UN coarse-level product data in Spanish.
lt-un-data-fine-industry-es	UN product data linked to industries in Spanish.
lt-un-data-fine-fine-fr	UN fine-level product data in French.
lt-un-data-fine-coarse-fr	UN coarse-level product data in French.
lt-un-data-fine-industry-fr	UN product data linked to industries in French.
lt-un-data-fine-fine-multi	UN fine-level product data in multiple languages.
lt-un-data-fine-coarse-multi	UN coarse-level product data in multiple languages.
lt-un-data-fine-industry-multi	UN product data linked to industries in multiple languages.

Table S-2: Model names and training data sources for various models in the `LinkTransformer` model zoo. Each of these models is on the Hugging Face hub and can be found by prefixing the organization name *dell-research-harvard* (for example, *dell-research-harvard/lt-wikidata-comp-multi*.). Training code can be found on our package Github repo and training configs containing the hyperparameters are available in the model repo on the Hugging Face Hub.

Model	Training Size	Validation Size	Test Size
lt-wikidata-comp-es	2252	359	345
lt-wikidata-comp-fr	6728	1034	1055
lt-wikidata-comp-ja	10120	1629	1616
lt-wikidata-comp-zh	5572	1087	1014
lt-wikidata-comp-de	10131	1514	1511
lt-wikidata-comp-en	28731	4132	4070
lt-wikidata-comp-multi	44660	149	149
lt-wikidata-comp-prod-ind-ja	1190	9950	9902
lt-un-data-fine-fine-en	1252	495	649
lt-un-data-fine-coarse-en	146	18	19
lt-un-data-fine-industry-en	146	18	19
lt-un-data-fine-fine-es	1252	274	310
lt-un-data-fine-coarse-es	146	18	19
lt-un-data-fine-industry-es	144	18	19
lt-un-data-fine-fine-fr	1185	210	255
lt-un-data-fine-coarse-fr	141	18	19
lt-un-data-fine-industry-fr	143	18	18
lt-un-data-fine-fine-multi	1252	210	255
lt-un-data-fine-coarse-multi	146	18	19
lt-un-data-fine-industry-multi	143	18	18

Table S-3: Model names and training, validation, and test sizes for various models in the `LinkTransformer` model zoo. The numbers correspond to the number of classes in each split - not the total number of examples. The data were split into test-train-val at the class level to avoid test set leakage.

Model	Edit Distance	SBERT	LT	OpenAI
Company Linkage				
lt-wikidata-comp-es	0.69	0.68	0.75	**0.81**
lt-wikidata-comp-fr	0.55	0.78	**0.84**	0.80
lt-wikidata-comp-ja	0.53	0.62	**0.70**	0.62
lt-wikidata-comp-zh	0.61	0.73	**0.80**	**0.80**
lt-wikidata-comp-de	0.63	0.69	**0.77**	**0.77**
lt-wikidata-comp-en	0.58	0.72	**0.74**	**0.74**
lt-wikidata-comp-multi	0.69	0.62	**0.80**	0.74
lt-wikidata-comp-prod-ind-ja	0.96	0.95	**0.99**	0.98
Fine Product Linkage				
lt-un-data-fine-fine-en	0.58	0.76	**0.83**	0.78
lt-un-data-fine-fine-es	0.57	0.67	**0.76**	0.68
lt-un-data-fine-fine-fr	0.52	0.67	**0.73**	0.68
lt-un-data-fine-fine-multi	0.52	0.57	**0.73**	0.68
Product to Industry Linkage				
lt-un-data-fine-industry-en	0.29	0.74	0.84	**0.95**
lt-un-data-fine-industry-es	0.41	**0.95**	0.89	0.89
lt-un-data-fine-industry-fr	0.19	**0.72**	0.67	0.61
lt-un-data-fine-industry-multi	0.19	**0.67**	**0.67**	0.61
Product Aggregation				
lt-un-data-fine-coarse-en	0.61	**0.95**	**0.95**	**0.95**
lt-un-data-fine-coarse-es	0.47	**0.95**	**0.95**	**0.95**
lt-un-data-fine-coarse-fr	0.53	0.89	**1.00**	0.95
lt-un-data-fine-coarse-multi	0.53	0.95	**1.00**	0.95

Table S-4: Performance of various embedding models. Performance is measured by retrieval accuracy at 1 - whether the nearest neighbor in terms of edit distance or cosine similarity is a "relevant" entity. *Company linkage* links company aliases together, *Fine Product Linkage* links products from different product classifications together, *Product to Industry Linkage* links products to their industry classifications, and *Product Aggregation* links a fine product to its coarser product classification. *LT* gives the performance of the trained LinkTransformer model specified in the Model Column (and found on HuggingFace). *Edit Distance* gives to linkage accuracy when using Levenshtein distance as the distance metric. *SBERT* gives linkage accuracy with the base off-the-shelf model used to tune the `LinkTransformer` model (as specified for each language in S-1). *OpenAI* gives linkage performance when using embeddings from the OpenAI embedding API (text-embedding-ada-002).

Dataset	Semantic Sim	Fine Tuned	Edit Distance	OpenAI ADA	LT UN/Wiki Model
mexicantrade4748	0.81	**0.89**	0.78	0.88	0.88
historicjapanesecompanies	0.66	0.79	0.29	**0.83**	0.80

Table S-5: Historical Linking. We examine the base semantic similarity model off-the-shelf, a fine-tuned `LinkTransformer` version, Levenshtein edit distance on the tariff description or company name, OpenAI embeddings and a pre-trained LinkTransformer model - *lt-wikidata-comp-prod-ind-ja* mexicantrade4748 and *lt-un-data-fine-fine-multi* for historicjapanesecompanies. The table reports top-1 accuracy.

Type	Dataset	Domain	Size	# Pos.	# Attr.	Ours (ZS)	Ours (FT)	Magellan	Deep matcher	Ditto	REMS
Structured	BeerAdvo-RateBeer	beer	450	68	4	82.35	87.5	78.8	72.7	84.59	96.65
	iTunes-Amazon1	music	539	132	8	70	80	91.2	88.5	92.28	98.18
	Fodors-Zagats	restaurant	946	110	6	88	93	100	100	98.14	100
	DBLP-ACM1	citation	12,363	2,220	4	90	97.5	98.4	98.4	98.96	98.18
	DBLP-Scholar1	citation	28,707	5,347	4	76	91.4	92.3	94.7	95.6	91.74
	Amazon-Google	software	11,460	1,167	3	39.4	68	49.1	69.3	74.1	65.3
	Walmart-Amazon1	electronics	10,242	962	5	28	69	71.9	67.6	85.81	71.34
Textual	Abt-Buy	product	9,575	1,028	3	33.6	78.4	33	55	88.85	67.4
	Company	company	1,12,632	28,200	1	74.07	88	79.8	92.7	41.00	80.73
Dirty	iTunes-Amazon2	music	539	132	8	74	81.3	46.8	79.4	92.92	94.74
	DBLP-ACM2	citation	12,363	2,220	4	79.6	97.2	91.9	98.1	98.92	98.19
	DBLP-Scholar2	citation	28,707	5,347	4	75	91.2	82.5	93.8	95.44	91.76
	Walmart-Amazon2	electronics	10,242	962	5	25	65	37.4	53.8	82.56	65.74

Table S-6: Benchmarks. ZS is `LinkTransformer` models zero-shot and FT is `LinkTransformer` models fine-tuned on the benchmark. The remaining columns report comparisons. The metric is F1, as these datasets frame linkage as a binary classification problem.

References

Ran Abramitzky, Leah Boustan, Katherine Eriksson, James Feigenbaum, and Santiago Pérez. 2021. Automated linking of historical data. *Journal of Economic Literature*, 59(3):865–918.

Abhishek Arora, Xinmei Yang, Shao Yu Jheng, and Melissa Dell. 2023. Linking representations with multimodal contrastive learning. *arXiv preprint arXiv:2304.03464*.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Olivier Binette and Rebecca C Steorts. 2022. (almost) all of entity resolution. *Science Advances*, 8(12):eabi8021.

Ursin Brunner and Kurt Stockinger. 2020. Entity matching with transformer architectures-a step forward in data integration. In *23rd International Conference on Extending Database Technology, Copenhagen, 30 March-2 April 2020*, pages 463–473. OpenProceedings.

Sanjib Das, A Doan, C Gokhale Psgc, Pradap Konda, Yash Govind, and Derek Paulsen. 2015. The magellan data repository.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*.

Forest Gregg and Derek Eder. 2022. dedupe.

Benjamin Hsu and Graham Horwood. 2022. Contrastive representation learning for cross-document coreference resolution of events and entities. *arXiv preprint arXiv:2205.11438*.

Jinji Koshinjo. 1954. *Nihon shokuinrokj*. Jinji Koshinjo.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Salil Rajeev Joshi, Arpan Somani, and Shourya Roy. 2021. Relink: Complete-link industrial record linkage over hybrid feature spaces. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2625–2636. IEEE.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment*, 3(1-2):484–493.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Yuliang Li, Jinfeng Li, Yoshi Suhara, AnHai Doan, and Wang-Chiew Tan. 2023. Effective entity matching with transformers. *The VLDB Journal*, pages 1–21.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584*.

Ralph Peeters and Christian Bizer. 2023. Using chatgpt for entity matching. *arXiv preprint arXiv:2305.03423*.

Anna Primpeli, Ralph Peeters, and Christian Bizer. 2019. The wdc training dataset and gold standard for large-scale product matching. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 381–386.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Secretaria de Economía de Mexico. 1948. Ajuste de las fracciones de la tarifa arancelaria que rigieron hasta el año de 1947 con las de la tarifa que entró en vigor por decreto de fecha 13 de diciembre del mismo año y se consideraron a partir de 1948. In *Anuario Estadístico del Comercio Exterior de los Estados Unidos Mexicanos*. Gobierno de Mexico.

Emily Silcock, Luca D'Amico-Wong, Jinglin Yang, and Melissa Dell. 2023. Noise-robust de-duplication at scale. *International Conference on Learning Representations*.

Jiawei Tang, Yifei Zuo, Lei Cao, and Samuel Madden. 2022. Generic entity resolution models. In *NeurIPS 2022 First Table Representation Workshop*.

Teikoku Koshinjo. 1957. *Teikoku Ginko Kaisha Yoroku*. Teikoku Koshinjo.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, volume 119, pages 9929–9939. PMLR.

Ross Wightman. 2019. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Xinmei Yang, Abhishek Arora, Shao-Yu Jheng, and Melissa Dell. 2023. Quantifying character similarity with vision transformers. *arXiv preprint arXiv:2305.14672*.

Huchen Zhou, Wenfeng Huang, Mohan Li, and Yulin Lai. 2022. Relation-aware entity matching using sentence-bert. *Computers, Materials & Continua*, 71(1).