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

Many computational analyses require linking information across noisy text datasets. While large language models (LLMs) offer significant promise, approximate string matching in popular statistical softwares such as R and Stata remain predominant in academic applications. These packages have simple interfaces and can be easily extended to a diversity of languages and settings, and for academic applications, ease-of-use and extensibility are essential. In contrast, packages for record linkage with LLMs require significant familiarity with deep learning frameworks and often focus on applications of commercial value in English. The open-source package `LinkTransformer` aims to bridge this gap by providing an end-to-end software for performing record linkage and other data cleaning tasks with transformer LLMs, treating linkage as a text retrieval problem. At its core is an off-the-shelf toolkit for applying transformer models to record linkage. `LinkTransformer` contains a rich repository of pre-trained models for multiple languages and supports easy integration of any transformer language model from Hugging Face or OpenAI, providing the extensibility required for many scholarly applications. Its APIs also perform common data processing tasks, *e.g.*, aggregation, noisy de-duplication, and translation-free cross-lingual linkage. `LinkTransformer` contains comprehensive tools for efficient model tuning, allowing for highly customized applications, and users can easily contribute their custom-trained models to its model hub to ensure reproducibility. Using a novel benchmark dataset geared towards academic applications, we show that `LinkTransformer`- with both custom models and Hugging Face or OpenAI models off-the-shelf - outperforms string matching by a wide margin. By combining transformer LMs with intuitive APIs, `LinkTransformer` aims to democratize these performance gains for those who lack familiarity 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 in social science applications, with rule-based methods continuing to overwhelmingly predominate (*e.g.*, see reviews by Binette and Steorts (2022); Abramitzky et al. (2021)). In particular, researchers commonly employ string-based matching tools available in statistical software packages such as R or Stata.

In academic applications, extensibility to a diversity of human societies (historic and present) and ease of use for those not familiar with deep learning are essential. String matching algorithms in widely used statistical packages meet these requirements because 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 scholarly 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

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 221–231

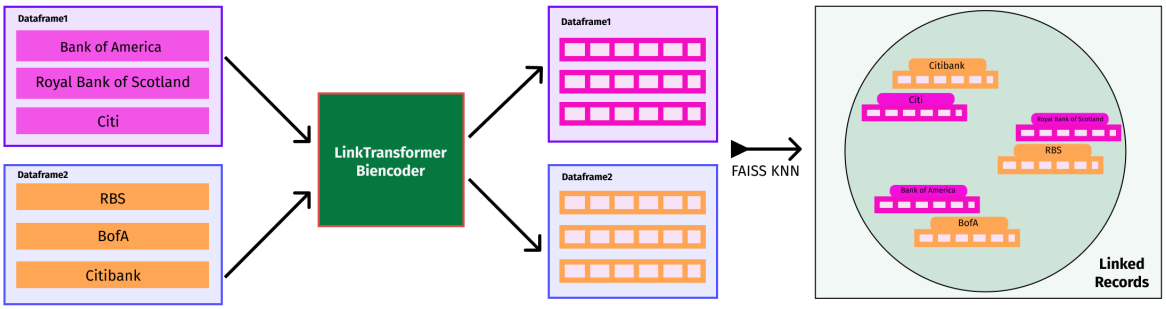

Figure 1: **Architecture.** This figure shows the `LinkTransformer` architecture for record linkage.

who lack extensive coding experience.

`LinkTransformer` integrates:

1. An off-the-shelf toolkit for applying transformer models to record linkage
2. A rich repository of pre-trained models, supporting multiple applications and languages and evaluated on novel social science-oriented benchmarks
3. Easy integration of any Hugging Face or OpenAI transformer LLM, for extensibility
4. APIs to support common data processing tasks: aggregation, de-duplication, classification, and translation-free cross-lingual linkage
5. Comprehensive tools for model tuning
6. Easy sharing to the `LinkTransformer` model hub, as reproducibility is essential for academic applications

While transfer learning can facilitate strong off-the-shelf performance, heterogeneity in how out-of-domain applications are from LLM training corpora - combined with settings that demand extremely high accuracy - create many scenarios where custom training may be needed. `LinkTransformer` makes it straightforward for users to tune their own customized models.

`LinkTransformer` performs well on challenging record linkage applications. It is equally applicable to 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, as there are many noisily measured fields (*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.

`LinkTransformer` has a GNU General Public License and is being actively maintained. A demo is available at `https://youtu.be/hFrh8k1pukI`. More resources are available on our package website `https://linktransformer.github.io/`.

This study is organized as follows: Section 2 provides an overview of related work. The core `LinkTransformer` library is described in Section 3. Section 4 evaluates `LinkTransformer` performance on various use cases, and Section 5 considers ethics.

2 Relation to the Existing Literature

The record linkage literature is sprawling - with large literatures in quantitative social science (particularly economics), statistics, computer science, and industry applications. These literatures are highly disjoint, taking very different approaches and even using different terms (record linkage, entity resolution/matching, approximate dictionary matching, etc.) to refer to the same task. A 2022 interdisciplinary *Science Advances* review, "(Almost) All of Entity Resolution" (Binette and Steorts, 2022), concludes that deep neural models are unlikely to be applicable to record linkage using structured data. It argues that training datasets are small and there is not much to be gained from LLMs since text fields are often short. Yet there is an active literature on e-commerce applications that underscores the utility of LLMs for linking structured datasets, even when text fields are short. Bench-

marks in this literature (e.g., Köpcke et al. (2010); Das et al. (2015); Primpeli et al. (2019)) focus on high resource commercial applications in English, 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 Zagat. Recent studies have used masked language models (Li et al., 2020; Joshi et al., 2021; Brunner and Stockinger, 2020; Zhou et al., 2022), GPT (Peeters and Bizer, 2023; Tang et al., 2022), or both, significantly outperforming static word embedding and other older linkage methods.

The siloed nature of the literature is reflected in softwares. The main existing package for record linkage with LLMs is Ditto (Li et al., 2020), which implements Li et al. (2023). It requires significant programming expertise to deploy. While this is appropriate for an e-commerce target audience - where data scientists predominate - the technical expertise required and the lack of pre-trained models targeted to multilingual social science applications has likely hindered further takeup. Moreover, most of the literature examining record linkage with LLMs poses record linkage as a classification task (Barlaug and Gulla, 2021), 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). In the social sciences, string matching with statistical packages predominates.

`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 bi-encoder applications with unstructured texts, *e.g.*, passage retrieval (Karpukhin et al., 2020), entity disambiguation (Wu et al., 2019), and entity coreference resolution (Hsu and Horwood, 2022). The knn retrieval structure of `LinkTransformer` also supports noisy de-duplication, a closely related task that finds noisily duplicated observations within a dataset, following the methods developed in Silcock et al. (2023).

`LinkTransformer` departs from much of the literature in utilizing LLMs trained for semantic similarity, combined with a supervised contrastive loss (Khosla et al., 2020). Off-the-shelf LLMs such as BERT have anisotropic geometries (Ethayarajh, 2019), which makes them unsuitable off-the-shelf for metric learning problems like `LinkTransformer` nearest neighbor retrieval. Contrastive training for semantic similarity reduces anisotropy, improving alignment between semantically similar pairs to be linked and improving sentence embeddings (Wang and Isola, 2020; Reimers and Gurevych, 2019). `LinkTransformer` builds closely upon the contrastively trained Sentence BERT (Reimers and Gurevych, 2019), whose semantic similarity library inspired many of the features in `LinkTransformer`.

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 minimal coding required. 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.

```
#pip install linktransformer
import linktransformer as lt
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)
```

We recommend that users new to LLMs deploy the package using a cloud service optimized for deep learning to avoid the need to resolve dependencies, and our tutorials use Colab.

In addition to supporting Hugging Face and OpenAI models, `LinkTransformer` provides pre-trained models, currently encompassing six languages (English, Chinese, French, German, Japanese, and Spanish) plus a multilingual model.

These models are trained and evaluated using novel datasets that reflect common record linkage tasks in quantitative social science:

1. **Firm aliases**: these are drawn from Wikidata for 6 languages. Firm alias models learn to recognize the different ways that firm names are written and abbreviated.

2. **Homogenized industry and product names:** These are drawn from the United Nations economic classification schedules (International Standard Industrial Classification, Standard International Trade Classification, and Central Product Classification), that map different country classifications to an international standard. We include models trained on these for 3 official UN languages.

3. **Historical datasets**: linked product-level Mexican tariff schedules from 1947 and 1948, and a dataset linking 1950s Japanese firms across noisy text databases. Historical data are central to better understanding economic and social processes; for example, these datasets could be used to elucidate the political determinants of tariff policy or the role of supply chain linkages in Japan's spectacular 20th century growth performance.

We also provide models for the standard industry benchmarks.

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

The `LinkTransformer` API supports a plethora of other features that are frequently integrated into data analysis pipelines. These include:

Aggregation: Data processing often requires the aggregation of fine descriptions into coarser categories, that are consistent across datasets and 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). `lt.aggregate_rows` performs this task, 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_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 SLINK, DBSCAN, HDBSCAN, and agglomerative clustering.

Cross-lingual linkage: Analyses spanning multiple countries often require cross-lingual linkage. Machine translation followed by string matching methods tend to perform very poorly, necessitating costly hand linking. `LinkTransformer` users can bypass translation by using multilingual transformer models.

Text Classification: While Hugging Face provides an accessible API, text classification can still be challenging for users who haven't been exposed to NLP libraries. Our API requires only one line of code to use a classification model on Hugging Face or the ChatGPT (3.5 and 4) API to classify texts.

Notebooks and tutorials outline the use of these functionalities on toy datasets.[1] We also have a tutorial to help those who are less familiar with language models to select ones that fit their use

[1] https://linktransformer.github.io/

case. More detailed information and additional features can be found in the online documentation.[2]

3.2 Customized Model Training

Record Linkage

Record linkage tasks are highly diverse and may demand very high accuracy; hence, fine-tuning on target datasets may be necessary. `LinkTransformer` supports easy model training, which can be initialized using any Hugging Face transformer model.

Training data are expected in a *pandas* data frame, removing entry barriers for the typical social science user. 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-model",
        data="df1.csv",
        left_col_names=["left_var"],
        right_col_names=['right_var'],
        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.

Classification

We added classification at the request of `LinkTransformer` users. Users can train custom models with a single line of code, using training data in the form of a data frame. They simply specify the on columns containing the text and a column for annotations, `label_col_name`. We have helpful guides on our website to allow users to effectively tune hyperparameters.

Since this function wraps around the *Trainer* class from Hugging Face, it can make use of multiple GPUs. `training_args` allow an advanced user to fully customize the *Trainer* by providing arguments with the same format as Hugging Face's *TrainingArguments* class.

3.3 User Contributions

`LinkTransformer` aims to promote reusability and reproducibility, which are central to academic applications. 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.

4 Applications

The LLMs in the `LinkTransformer` model zoo excel at a variety of tasks. Table 1 evaluates performance linking Wikidata firm aliases (panel A), linking product descriptions from different countries' classification schemes (panel B), linking products to their industries (panel C), and aggregating fine product descriptions to coarser descriptions (panel D). It compares the accuracy of Levenshtein edit distance matching (Levenshtein et al., 1966), popular off-the-shelf semantic similarity models from Hugging Face (see Appendix Table A-1 for a listing of models used), OpenAI embeddings (the better of `text-embedding-3-small` and `text-embedding-3-large`, which outperformed Ada embeddings), and `LinkTransformer` tuned models. The supplementary materials describe the models and training datasets in detail.

As expected, custom-tuned models typically achieve the best performance, with off-the-shelf models still outperforming edit distance matching, typically by a wide margin. The custom-trained models are often plausibly achieving human-level accuracy, as cases that they get wrong are often impossible to resolve from the information provided, *e.g.,* in cases where a firm is referred to by two completely disparate acronyms.

Second, we examine historical applications, which are central to understanding long-run phe-

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

Model	Edit Distance	SBERT	LT	OpenAI
Panel A: Company Linkage				
lt-wikidata-comp-fr	0.43	0.74	**0.81**	0.75
lt-wikidata-comp-ja	0.51	0.61	**0.70**	0.63
lt-wikidata-comp-zh	0.66	0.77	**0.83**	0.82
lt-wikidata-comp-de	0.51	0.66	**0.76**	0.71
lt-wikidata-comp-es	0.62	0.68	0.75	**0.82**
lt-wikidata-comp-en	0.36	0.60	**0.70**	0.64
lt-wikidata-comp-multi	0.55	0.69	**0.83**	0.77
lt-wikidata-comp-prod-ind-ja	0.48	0.97	**0.99**	0.98
Panel B: Fine Product Linkage				
lt-un-data-fine-fine-en	0.64	0.82	**0.87**	0.84
lt-un-data-fine-fine-es	0.42	0.68	**0.80**	0.72
lt-un-data-fine-fine-fr	0.45	0.71	**0.75**	0.72
lt-un-data-fine-fine-multi	0.54	0.79	**0.84**	0.77
Panel C: Product to Industry Linkage				
lt-un-data-fine-industry-en	0.18	0.81	**0.80**	0.73
lt-un-data-fine-industry-es	0.18	0.67	**0.72**	0.64
lt-un-data-fine-industry-fr	0.14	0.56	**0.72**	0.55
lt-un-data-fine-industry-multi	0.10	0.69	**0.78**	0.75
Panel D: Product Aggregation				
lt-un-data-fine-coarse-en	0.27	0.76	0.85	**0.86**
lt-un-data-fine-coarse-es	0.24	0.75	**0.80**	0.7
lt-un-data-fine-coarse-fr	0.24	0.74	**0.77**	0.69
lt-un-data-fine-coarse-multi	0.22	0.6	**0.64**	0.62

Table 1: Performance of various embedding models, measured by top-1 retrieval accuracy. *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. *Edit Distance* gives linkage accuracy when using Levenshtein distance, and *SBERT* when using semantic similarity models off-the-shelf (See Table A-1). *OpenAI* gives the best linkage performance when using embeddings from OpenAI embedding models.

nomena like economic growth or social mobility and typically lack unique identifiers for linkage. First, we link two tariff schedules published by the Mexican government in the 1940s (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 in a rare crosswalk published by the government (typically, homogenized crosswalks do not exist). 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 Open AI embeddings. All transformer models widely outperform edit distance. While there are considerable debates on the role that trade policies have played in long-run development, empirical evidence is limited largely due to the considerable challenges of homogenizing tariff schedules across time. Language model

Dataset	Semantic Sim	Fine Tuned	Edit Distance	OpenAI ADA	LT UN/Wiki Model
mexicantrade4748	0.75	**0.88**	0.70	0.83	0.80
historicjapan	0.69	**0.91**	0.27	0.86	0.74

Table 2: 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. The table reports top-1 accuracy.

linking offers the opportunitiy to bring novel quantitative evidence to this important question.

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. We concatenate information on the firm's name, prefecture, major products, shareholders, and banks. These variables contain OCR noise and the information included varies, *e.g.* in terms of how a firm's products are described, which shareholders are included, etc. This makes rule-based methods quite brittle, whereas the custom-tuned model links 91% of firms correctly.

In the supplemental materials, we examine the various e-commerce and industry benchmarks that prevail in this literature. We use 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 used with `LinkTransformer` if on Hugging Face) and sometimes outperforming other models.

When OCR errors are severe, too much information may have been destroyed to achieve the desired accuracy with the garbled texts. A multimodal matching framework (Arora et al., 2023) that uses aligned language and vision transformers to incorporate the original image crops or a matching framework that incorporates character visual similarity (Yang et al., 2023) - as OCR confuses visually similar characters - may be required. Vision and multimodal linking support will be incorporated into future releases of `LinkTransformer`.

5 Ethics Statement

`LinkTransformer` is ethically sound. It is built using public domain training data. Because it is built

upon transformer language models, it will not be suitable for lower resource languages that lack pre-trained LLMs. However, it can utilize any Hugging Face or OpenAI embedding model and hence will be extensible as the low-resource transformer literature expands to lower resource settings, as long as relevant embedding models are posted on Hugging Face or made available commercially by OpenAI.

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*.

Nils Barlaug and Jon Atle Gulla. 2021. Neural networks for entity matching: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(3):1–37.

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.

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).

A Supplementary Materials

A.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) and Online Contrastive loss with default hyperparameters as the training objective depending upon the structure of the training dataset (as specified in Table A-4). The implementation for the losses 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 is available on our repository, which also contains links to our training data.

A.2 Datasets and Results

Table A-1 lists the base sentence transformer models that we used to initialize the custom `LinkTransformer` models. Table A-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 A-3 describes the train-val-test splits for each of these datasets. Table A-5 reports results on standard industry and e-commerce benchmarks for record linkage.

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-manet-base-v2

Table A-1: We used the above sentence-transformers models for different languages 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 A-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	16511	924	932
lt-wikidata-comp-fr	42475	2431	2486
lt-wikidata-comp-ja	35923	2035	2054
lt-wikidata-comp-zh	26224	1510	1513
lt-wikidata-comp-de	42647	2383	2377
lt-wikidata-comp-en	133557	7685	7648
lt-wikidata-comp-multi	381820	28682	30532
lt-wikidata-comp-prod-ind-ja	3647	149	149
lt-un-data-fine-fine-en	9545	569	587
lt-un-data-fine-coarse-en	8059	1399	614
lt-un-data-fine-industry-en	8644	977	474
lt-un-data-fine-fine-es	5462	289	305
lt-un-data-fine-coarse-es	4326	552	389
lt-un-data-fine-industry-es	4134	622	530
lt-un-data-fine-fine-fr	1185	249	204
lt-un-data-fine-coarse-fr	3191	546	261
lt-un-data-fine-industry-fr	3219	501	302
lt-un-data-fine-fine-multi	19311	374	443
lt-un-data-fine-coarse-multi	17939	529	911
lt-un-data-fine-industry-multi	16528	1974	888
lt-mexicantrade4748	5348	466	477
lt-historicjapan	982	50	55

Table A-3: Model names and training, validation, and test sizes for various models in the `LinkTransformer` model zoo. The training size corresponds to the number of samples (or pairs for training with online contrastive loss) in the split. Validation and Test size correspond to the number of 'queries' for models evaluated on the retrieval task and to positive pairs for models evaluated on the paired classification task (For *historicjapan*). The data were split into test-train-val at the class level to avoid test set leakage whenever possible.

Dataset	Model	Loss
Structured_Amazon-Google	multi-qa-mpnet-base-dot-v1	supcon
Structured_Beer	bge-large-en-v1.5	onlinecontrastive
Structured_DBLP-ACM	bge-large-en-v1.5	onlinecontrastive
Structured_DBLP-GoogleScholar	bge-large-en-v1.5	onlinecontrastive
Structured_iTunes-Amazon	bge-large-en-v1.5	onlinecontrastive
Structured_Walmart-Amazon	bge-large-en-v1.5	supcon
Structured_Fodors-Zagats	bge-large-en-v1.5	supcon
Dirty_DBLP-ACM	bge-large-zh-v1.5	supcon
Dirty_DBLP-GoogleScholar	bge-large-zh-v1.5	supcon
Dirty_iTunes-Amazon	all-mpnet-base-v2	supcon
Dirty_Walmart-Amazon	bge-large-zh-v1.5	supcon
Textual_Abt-Buy	multi-qa-mpnet-base-dot-v1	onlinecontrastive

Table A-4: Base models and Loss functions used for training of industrial benchmarks. Other hyperparameters that were constant across all of these experiments - learning rate (2e-5) , batch size (128), linear warmup of a 100% (reaching the maximum learning rate). All models were run for 100 epochs and the checkpoint was selected on the basis of test F1 on validation set. Since labels (and also negatives) were also in the dataset, validation was done by pairwise classification.

Type	Dataset	Domain	Size	# Pos.	# Attr.	Ours (ZS)	Ours (FT)	Magellan	Deep matcher	Ditto	REMS
Structured	BeerAdvo-RateBeer	beer	450	68	4	83.38	90.32	78.8	72.7	84.59	96.65
	iTunes-Amazon1	music	539	132	8	60.6	90	91.2	88.5	92.28	98.18
	Fodors-Zagats	restaurant	946	110	6	75	98	100	100	98.14	100
	DBLP-ACM1	citation	12,363	2,220	4	95	98	98.4	98.4	98.96	98.18
	DBLP-Scholar1	citation	28,707	5,347	4	80	92	92.3	94.7	95.6	91.74
	Amazon-Google	software	11,460	1,167	3	47.1	74	49.1	69.3	74.1	65.3
	Walmart-Amazon1	electronics	10,242	962	5	45	73.8	71.9	67.6	85.81	71.34
Textual	Abt-Buy	product	9,575	1,028	3	28.8	84	33	55	88.85	67.4
	Company	company	1,12,632	28,200	1	74.07	88.00	79.8	92.7	41.00	80.73
Dirty	iTunes-Amazon2	music	539	132	8	68.8	84	46.8	79.4	92.92	94.74
	DBLP-ACM2	citation	12,363	2,220	4	89.8	98	91.9	98.1	98.92	98.19
	DBLP-Scholar2	citation	28,707	5,347	4	87.5	92.6	82.5	93.8	95.44	91.76
	Walmart-Amazon2	electronics	10,242	962	5	45	71	37.4	53.8	82.56	65.74

Table A-5: 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.