Entity Linking of Sound Recordings and Compositions with Pre-trained Language Models

Nikiforos Katakis and Pantelis Vikatos
Orfium Research, Kallirois 103, 117 45, Athens, Greece

Keywords: Industrial Entity Matching, Deep Learning, Language Models.

Abstract: In this paper, we present a Deep Learning (DL) approach to tackle a real-world, large-scale music entity matching task. The quality of data, the lack of necessary information, and the absence of unique identifiers affect the effectiveness of entity matching and pose many challenges to the matching process. We propose an efficient matching method for linking recordings to their compositions through metadata using pre-trained language models. We represent each entity as a vector and estimate the similarity between vectors for a pair of entities. Our experiments show that an application of language models such as BERT, DistilBERT or ALBERT to large text corpora significantly improves the matching quality at an industrial level. We created a human-annotated dataset with sound recordings and composition pairs obtained from music usage logs and publishers, respectively. The proposed language model achieves 95% precision and reaches 96.5% recall which is a high performance on this challenging task.

1 INTRODUCTION

The rise of digital distribution of music in recent decades has led artists, writers & producers to take a greater interest in recognising the use and rights of their works on digital platforms and streaming services (Skog et al., 2018). Collective Management Organisations (CMOs), whose goal is to pay royalties to various parties based on music usage, rely on usage log metadata to match music with publishers' repertoire and available catalogues (Panda and Patel, 2012). Both record record labels and independent artists provide a unique identifier, the International Standard Recording Code (ISRC), which is assigned to a particular sound recording by an artist or band. At the same time, worldwide publishers list a unique code, the International Standard Musical Work Code (ISWC), assigned to a particular musical work or songwriter's composition in their composition list. However, if one or both identifiers are missing, the linking of entities can only be done through the available metadata. The quality of data, the lack of necessary information, and the absence of unique identifiers pose numerous challenges to the matching process. The task of linking data from heterogeneous databases, to which there is no unique and primary key linking the data, reduces to the problem of Entity Resolution (ER) (Christophides et al., 2019). Several studies propose the use of machine learning methods to accomplish ER tasks (Bilenko and Mooney, 2003; Cohen and Richman, 2002; Sarawagi and Bhamidipaty, 2002) where the decision about a matched or mismatched pair is derived from the model (e.g., decision tree or SVM) using extensive preprocessing and pairwise extraction of linguistic features. Other approaches provide a set of rules, e.g. Ruled-Based (RB); from experts to collect guidelines for matching and decision thresholds (Singla and Domingos, 2006; Konda et al., 2016). Deep Learning (DL) methods have been presented for a variety of prediction tasks such as natural language processing (NLP) (Goldberg, 2016; Hirschberg and Manning, 2015) and computer vision (CV) (He et al., 2016; Krizhevsky et al., 2012). Recent studies provide solutions using DL entity resolution models through two approaches. The first approach focuses on representing the entity as a vector and estimating the similarity between vectors, known as Representation Based Method (Ebraheem et al., 2018). The second approach, i.e., Compare-Aggregate Based Method, uses two-step processing in which pairwise attribute values are compared to obtain multiple matching vectors, and the extracted matching vectors are aggregated to produce the final similarity score (Mudgal et al., 2018).

Our paper is categorized as Representation Based Methods. Given a list of candidate pairs with meta-

Katakis, N. and Vikatos, P.
Entity Linking of Sound Recordings and Compositions with Pre-trained Language Models.
DOI: 10.5220/0010713900003058
In *Proceedings of the 17th International Conference on Web Information Systems and Technologies (WEBIST 2021)*, pages 474-481
ISBN: 978-989-758-536-4; ISSN: 2184-3252

Table 1: Linguistic variations of entities in song title and artist names.

Issue	**Example**
Misspellings	Led Zepplin = Led Zeppelin
Numbers	The 3 Tenors = The Three Tenors
Stylized Names	NSync = 'N Sync
Missing Terms	The Sex Pistols = Sex Pistols
Acronyms	B.D.P. = BDP = Boogie Down Productions
Initials	J.S. Bach = Johann Sebastian Bach
Lead Performers	Sting & The Police = The Police
Misencodings	©PªNÛ = Jay Chou
Transliterations	Jay Chow = Jay Chou
Translated Names	Chou Jie Lun = Jay Chou
Legal Changes	Yaz (EN-US) = Yazoo.
Localization	Tchaikovsky (EN) = Чайковский (RU)

data of sound recordings and compositions, Entity Resolution generates a list of matched recommendations with a score reflecting confidence in the match (Christophides et al., 2019). Challenges include the lack of necessary title, artist, and composer information. In addition, redundant information can lead to missing matches. Linguistic features such as misspellings, typos, acronyms, or polysemy are factors that make matching difficult. The current improvements in the language model lead to innovative approaches to solve the entity resolution problem. In this paper, we present an approach using pre-trained state-of-the-art language models for efficient entity resolution between pairs of recordings and compositions. The rest of the paper is structured as follows. Section 2 describes the data creation process which is separated into sub-modules. Section 3 mentions preliminaries of language models, detailed information about proposed DL models and overviews details of the implementation. Section 4 presents a reference to our experimental results and the discussion of our work. Finally, in Section 5, we discuss the strengths and limitations of our approach and we conclude with an outlook to future work.

2 DATASET DESCRIPTION

There is a wealth of publicly available datasets for entity resolution (Köpcke et al., 2010; Primpeli et al., 2019). However, there is a gap for a dataset containing recordings and corresponding composition pairs, which is the focus of this work. This section focuses on creating the dataset for training and evaluating Deep Learning models. We create this type of dataset by using the procedure of Figure 1. The proposed method of data creation is usually adopted by entity resolution tasks that aim to tame their quadratic complexity and scale them to large datasets (Papadakis et al., 2016).

Our goal is to create a dataset containing instances of accepted and rejected pairs of recordings and associated compositions. A list of recordings and a catalogue of musical compositions form the raw data and source material for candidate pair extraction. More precisely, one catalogue contains the basic metadata of a recording, i.e. title, artist and writer names. The other catalogue contains the titles of the compositions and the writer names. As for the writers, in Table 2 we define the categorization of the writers corresponding to the song rights holders.

2.1 Data Preprocessing

The first module of data creation corresponds to data preprocessing. Sound recording metadata is derived from music usage logs and may have multiple formats and versions. Figure 2 shows the diversity of the song *Hotel California by Eagles* in the raw data, which includes mixdowns of the original sound recordings such as remix, bass mix and workout mix. It also includes special live performances of the song and cover versions like karaoke and instrumental. It also adds details about the artist or band in the recording tile or uses redundant information such as "made famous by".

In summary, music metadata often contains linguistic variations of entities in song titles and artist names, as in Table 1, which complicates the retrieval and matching process. Preprocessing allows tokenization of titles and separation of artists and writers. It then removes stop words and punctuation that are irrelevant to the search and increase the indexing volume. Finally, it detects music terms, i.e. live performance, guitar version, etc., in the title and writer fields, which are redundant information for entity matching.

2.2 Indexing & Query Process

Given 2 lists of $|N|$ records and $|K|$ compositions, the total number of pairs is $|N| * |K|$. When the lists are very large, the number of candidates for each matching procedure is prohibitive. Therefore, the second step helps to extract the candidate pairs efficiently and reduce the time and space complexity. The module starts with indexing using inverted files of the terms of each composition sample to achieve efficient time performance and find similar compositions/documents. It also provides query processing where the values of the query attributes are checked for indexing and then searched to find the candidate records.

The catalog of compositions has $2,123,224$ unique cases and our list of available recordings is $23,332$. The best known technique to reduce the number of candidates is to introduce a blocking key in the process query module. More precisely, we introduce a blocking key representing (at least) one token match in the title, and we limit the upper bound of the block size based on the predefined threshold for the maximum number of results threshold K. We find relevant documents for each query using the probabilistic relevance framework of BM25 (Robertson and Zaragoza, 2009), which provides a ranked list of documents with scores in the range $[0-1]$. The number of results for each query is equal to the size of the list, so we restrict ourselves to the top 10 results that our experiments show to have high performance on pairwise completeness. We summarise all the rankings resulting from queries to produce a list of candidate pairs.

2.3 Voting & Consensual Annotations

The generated candidate pairs are fed to the annotation process, where human experts decide whether to accept or reject the proposed relationships. The quality of the dataset depends on the annotation and we use a voting procedure for the final decision. The decision criteria are derived from the knowledge about the music industry using the consensus of 3 domain experts. Pairs that do not meet the linkage criteria are removed from the final dataset.

The method used to create the dataset generated $33,727$ annotated pairs of recordings and compositions with $18,724$ and $15,003$ accepted and rejected pairs, respectively. The generated dataset contains $23,332$ and $16,463$ unique titles and artists in recordings, respectively. The number of unique composition titles is $16,784$ and the number of unique collaborations of composers provided by the dataset is $16,392$. Table 3 summarises the basic descriptive statistics for the generated dataset.

3 MODEL OVERVIEW

In this section, we present preliminary work on pretrained language models and describe in detail the architecture of our model.

3.1 Language Models

Language Models (LMs) use various statistical and probabilistic techniques to determine the probability of a particular sequence of words occurring in a sentence. They analyze text segments to provide a basis for their word prediction. Recently, LMs have been used in many natural language processing (NLP) tasks and have produced great results.

One of the most successful LM is BERT (Bidirectional Encoder Representations from Transformers), published by researchers at Google AI Language (Devlin et al., 2018). It has achieved top results on many NLP tasks, such as Question Answering, Natural Language Inference, etc (Jiang and de Marneffe, 2019; Qu et al., 2019). The biggest technical innovation of BERT is the application of the bidirectional training of Transformer, a popular attention model, to language modeling. This has enabled a deeper sense of language context and improved on the old technique of looking at a text sequence from left to right.

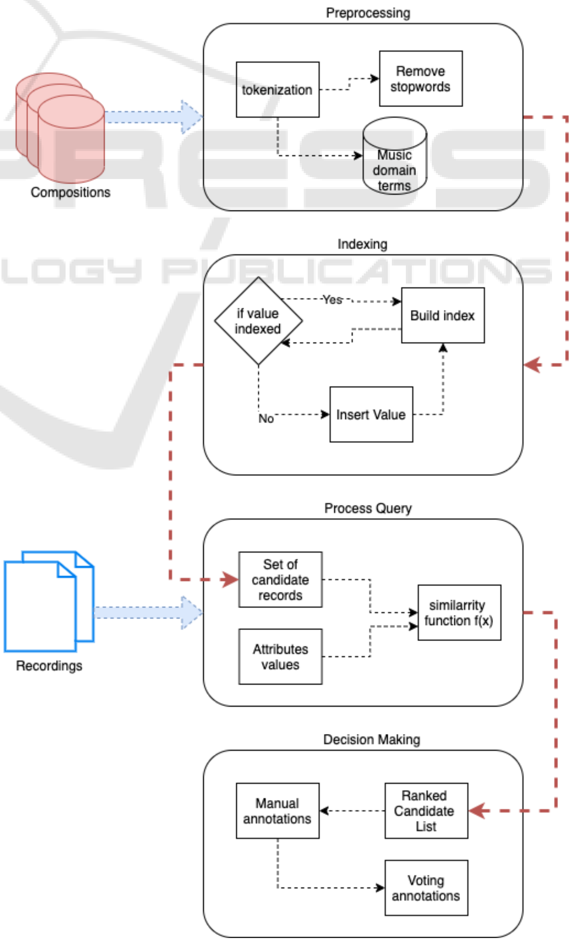


Figure 1: The architecture of the dataset creation process based on 4 independent modules.

Table 2: Sub-categories of writers in compositions.

Writer Type	Writer Name	Work
composer	Trent Reznor	Closer (Nine Inch Nails)
lyricist	Stephen Sondheim	Sweeney Todd: The Demon Barber of Fleet Street
librettist	Charles Jennens	Messiah, HWV 56
revised by	Igor Stravinsky	Pater Noster
translator	August Wilhelm Schlegel	Ständchen ("Horch, horch, die Lerche"), D. 889
reconstructed by	Simon Heighes	Markus-Passion, BWV 247

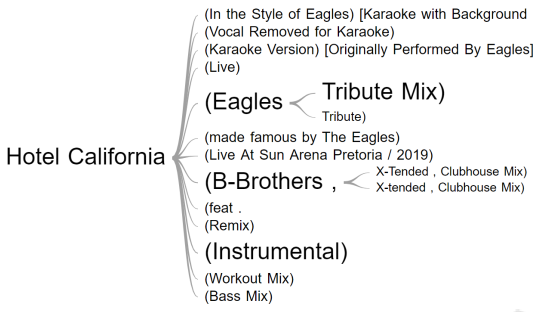

Hotel California
(In the Style of Eagles) [Karaoke with Background
(Vocal Removed for Karaoke)
(Karaoke Version) [Originally Performed By Eagles]
(Live)
(Eagles
Tribute Mix)
Tribute)
(made famous by The Eagles)
(Live At Sun Arena Pretoria / 2019)
(B-Brothers ,
X-Tended , Clubhouse Mix)
X-tended , Clubhouse Mix)
(feat .
(Remix)
(Instrumental)
(Workout Mix)
(Bass Mix)

Figure 2: Word Tree generated from song Hotel California.

Transformer (Vaswani et al., 2017) is an attentional mechanism that learns contextual relations between words in a text. It involves two separate mechanisms, an encoder that reads the text input and a decoder that makes a prediction for the task. Since BERT is a Language Model, only the encoder mechanism is used. The transformer encoder reads the entire word sequence of words at once and is therefore not directional. This feature allows the model to learn the context of a word based on the entire sequence.

3.2 Model Architecture

Recently, there are several approaches that address the matching problem using rules derived by human experts or by machine learning (Konda et al., 2016). Deep Learning for the problem EM has gained interest and achieved promising results (Ebraheem et al., 2018; Fu et al., 2019), mainly focusing on tailored RNN architectures and word embeddings. Our approach creates a simpler architecture based on fine-tuning pre-trained LMs. One of the advantages of pre-trained LMs is that they can be fine-tuned to better perform certain tasks. Study (Li et al., 2020) compares pre-trained language models with a set of entity matching tasks and presents the results of BERT, the distilled version, DistilBERT (Sanh et al., 2019) and ALBERT (Lan et al., 2019) as those with the highest performance. We adopt the above language models and add a fully connected layer and a softmax for output to address the problem as a binary classification and fine-tune the network, as Figure 3 shows

We also explore the approach of introducing domain knowledge as proposed by study (Li et al., 2020). In this approach, domain knowledge is injected into each entry in a preprocessing step using a pre-trained Named-Entity Recognition (NER) model. The concept behind this technique is that when humans try to identify whether a pair matches or not, they first search for text segments that contain the most relevant information and then make their final decision. We use the pre-trained NER model to search for useful entities in each entry, and enclose these entities with a type label.

Table 3: Descriptive statistics of the dataset.

	Attribute	Value
Recording	Titles	23,332
	Artists	16,423
	Writers Collaborations	16,784
Composition	Titles	16,620
	Writer Collaborations	16,392

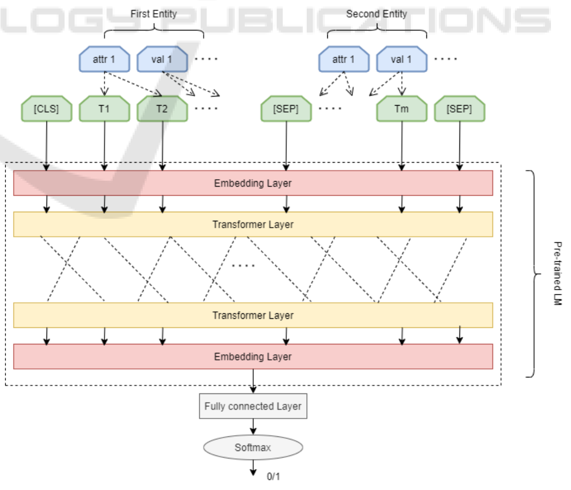


Figure 3: Overview of our Model's Architecture, adding a Fully Connected Layer to the Transformers based architecture of the Pre-Trained LMs.

The network is initialized with the weights of the pre-trained LM and then trained on our task-specific dataset for K epochs (K=20).

3.3 Input Serialization

One of the challenges of this approach is representing candidate pairs and converting them into a valid input to the LM. LMs require sequences of tokens (text) as input and we serialize the input as follows:

Each pair consists of two entries, which may be either recordings or compositions. For each of these k entries:

$$e = \{(attr_i, val_i)\}_{1<i<k} \quad (1)$$

If the entry is a recording, then the attributes are *Title*, *Artists*, and *Writers*. If the entry is a Composition, then the attributes are *Title* and *Writers*. In either case, the entry e is serialized as follows:

$$attr_{title} = [COL]title[VAL]val_{title} \quad (2)$$

$$attr_{writers} = [COL]title[VAL]val_{writers} \quad (3)$$

$$attr_{artists} = [COL]title[VAL]val_{artists} \quad (4)$$

$$\overline{e} ::= attr_{title} \cup attr_{writers} \cup attr_{artists} \quad (5)$$

where [COL] and [VAL] are tokens indicating the beginning of an attribute and the value of the attribute, respectively. The attribute *artist* is empty in the case of a composition.

In order to serialize a candidate pair of entries, we let,

$$\overline{(e, e')} = [CLS]\overline{e}[SEP]\overline{e'}[SEP], \quad (6)$$

where [CLS] is a special token for BERT to encode the pair sequence into a high-dimensional vector to transition to the fully connected layer and [SEP] is a token to indicate the end of an entry.

3.4 Implementation

The network was implemented in Python using the PyTorch framework[1] and the Transformers library[2]. The maximum sequence length was set to 256 and the learning rate started at 3e-5 and decreased linearly. We used a batch size of 32 and trained for 20 epochs, storing the model with the highest F1 score in the validation set. To run the experiments, we used AWS SageMaker[3] on a ml.p3.2xlarge (Tesla V100 GPU).

4 RESULTS

To get a better insight into the quality of the proposed models, we present a detailed description of the evaluation results. We train the variants of LMs on the same training dataset and evaluate the performance

[1]PyTorch: https://pytorch.org/
[2]Transformers: https://huggingface.co/transformers/
[3]AWS SageMaker: https://aws.amazon.com/sagemaker/

Table 4: Performance metrics of trained models.

Language Model	Evaluation Metrics			
	Precision	**Recall**	**F1**	**FPR**
BERT	**0.9546**	0.957	0.9558	**0.0628**
Distil-BERT	0.9499	**0.9649**	**0.9574**	0.0702
ALBERT	0.9524	0.9371	0.9447	0.0646

on a percentage of the dataset generated by the data process of Section 2. We compare the performance in this particular task for BERT, DistilBERT and ALBERT language model. We keep a detached part of the dataset as a test set representing the 10% of the initial set. We treat this problem as a binary classification and evaluate the performance of our model using the task-related metrics of precision, F1-score, recall, and false positive rate as defined below:

- *Precision* is measured as $prec = \frac{|TP|}{|TP|+|FP|}$ and is the fraction of classified matches that are true matches.
- *Recall* is measured as $rec = \frac{|TP|}{|TP|+|FN|}$ and is the proportion of true matches that are correctly classified.
- *F-measure* is the harmonic mean of precision and recall and is calculated as $F1 = 2 \times \left(\frac{prec \times rec}{prec + rec}\right)$.
- *False positive rate* is measured as $fpr = \frac{|FP|}{|TN|+|FP|}$.

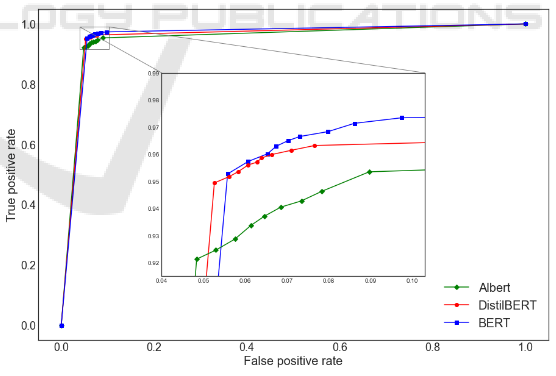


Figure 4: Comparison of the pre-trained language models through receiver operating characteristic curve as a binary classifier system which its discrimination threshold is varied.

The evaluation results show that the LMs achieve high performance and can handle this challenging task efficiently. Table 4 describes the evaluation metrics on the test set based on the pre-trained LM, which are used as the backbone of the network. The precision varies in a range of 0.949 – 0.954 and the model BERT performs better in precision than the others, reaching the highest value of 0.954. Also, the model

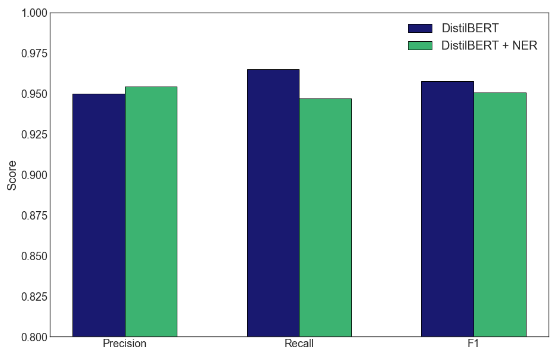


Figure 5: Comparison of the DistilBERT against DistilBERT + NER in precision, recall and F1.

BERT has the lowest false positive rate of 0.063. The recall varies in a range of 0.940 – 0.965 and DistilBERT achieves the highest value. DistilBERT achieves the best upper bound of F1 with 0.957. The experimental results with ALBERT show comparatively lower performance on the evaluation metrics. Moreover, in Figure 4, we show the ROC curve, which combines the true and false positive rates and provides a comparison factor for binary classification methods. BERT and DistilBERT have similar performance, while ALBERT lags in performance but still achieves high scores in precision and recall given the difficulty of the task. Considering that DistilBERT is a stripped down version of BERT that uses knowledge distilation and gives almost similar performance results, we conclude that DistilBERT is our best option.

We provide results for the scalability of the best model classifier, i.e. DistilBERT; as Figure 7 shows. We use three scaling steps of 20%, 50%, and 100%, each representing percentages of the training dataset. The scaling shows that even with only 20% of the total dataset, the model achieves an F1 score of almost 0.929 and recall reaches 0.945. Moreover, despite the small dataset, DistilBERT achieves a precision of 0.912, demonstrating the immense power and ability to efficiently predict pairwise linkage even with a really small training dataset. The success rate of our best model converges in the range of more than 50% and the improvements in precision and recall are in the range of 0.005 . A significant improvement is achieved by expanding the training dataset from 20% to 50%, which increases precision by 3.2% and recall by 1.2%.

We evaluate the approach of introducing domain knowledge using the Named-Entity Recognition (NER) model to search for useful entities in each entry, and we wrap these entities with a type label. A robust and well-known NER model with efficient results

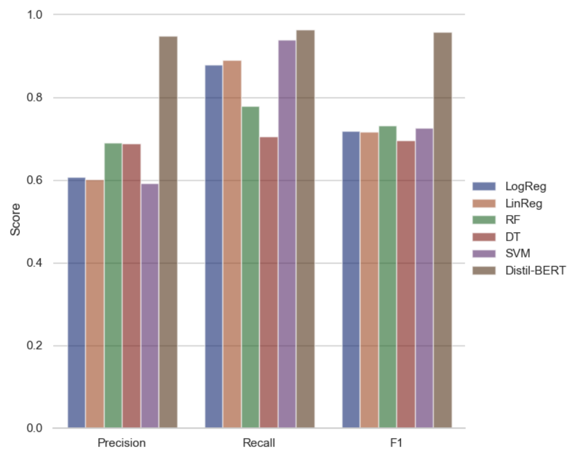


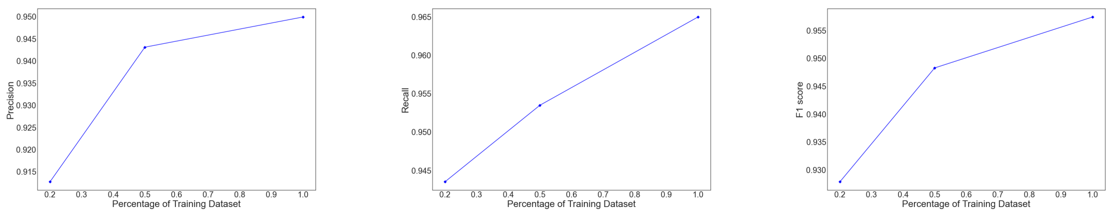
Figure 6: Comparison of the DistilBERT aganinst baseline ML models in precision, recall and F1.

is provided by spaCy[4]. We filter for specific types, e.g., persons, dates, works-of-art, organizations, replacing the original text with a new text in which special tokens are inserted around the text segments of each identified entity to reflect its type. These new tokens become signals to the self-attention mechanisms of the pre-trained LMs and ensure that the texts are better aligned for matching. Figure 5 compares the evaluation metrics of DistilBERT and DistilBERT+NER. The results show a slight improvement in precision and FPR, but a decrease in F1 score and recall. More specifically, DistilBERT+NER achieves 0.954, 0.946 precision and recall, respectively, and the false positive rate is 0.062.

We also provide insight into training time performance, as shown by Table 5. As expected, BERT is the slowest model, taking almost twice as much time to complete an epoch compared to the other 2 models. DistilBERT and ALBERT are close, but since ALBERT the metrics are a bit lower.

In addition, we compared the best LM to basic ML models using the public Python module py_entitymatching from AnHai Doan's group [5]. The module includes Entity Linking Workflows, to select the best learning-based matcher for the specific data, and is commonly used as a baseline for Entity Linking. The available learning-based matchers are. Linear Regression (LinReg), Logistic Regression (LogReg), Decision Tree (DT), Decision Tree (DT), Random Forest (RF), and Support Vector Machines (SVM). We compare the LM with all the above matchers with default settings. The module generates a feature vector using a combination of well known fuzzy matching algorithms like Levenshtein Simi-

[4]spaCy NER: https://spacy.io/api/entityrecognizer
[5]http://anhaidgroup.github.io/py_entitymatching



(a) Precision (b) Recall (c) F1 metric

Figure 7: The impact of scaling in performance metrics for DistilBERT model.

Table 5: Training time performance of each model to complete a whole epoch.

Pre-Trained Model	Training Time
BERT	5min 26sec
DistilBERT	3min 41sec
ALBERT	2min 34sec

larity, Monge-Elkan, Needleman-Wunsch & Smith-Waterman algorithm. Figure 6 shows the comparison of the base models with the LM. The results show that the approaches of ML fall short of the proposed LM. This shows that the traditional data mining approaches of technical features and training of a model are not sufficient to accomplish this task EL. More specifically, RF represents the best ML model, however, the precision is much lower compared to DistilBert, which shows the effectiveness of the language model. We should note that preprocessing can have a large impact on the performance of ML models, but it is not a necessary step for a pretrained LM.

5 CONCLUSIONS & FUTURE WORK

In this paper, we introduce the use of pre-trained language models to tackle the challenging task of entity linking sound recordings with related compositions utilizing metadata pairs.

The main points of our contribution can be summarized in the following sentences:

- We describe a fast and trusted dataset creation process using the sub-modules of preprocessing, indexing, candidate reduction and consensual labelling.
- We train and evaluate state of the art pre-trained Language models providing performance metrics of their predictive ability, scaling and training consumption.

We benefit from pre-trained LMs based on the Transformer technology and we fine-tune them for our task. The results show that it performs extremely well on our dataset, even with a small amount of data. The best model reaches 95% and 96.5% precision and recall respectively, which constitutes a high performance in this challenging task. The huge power of pre-trained LMs is shown when the results remain competitive even with only 10% of the training dataset. LMs contribute to the avoidance of heavy preprocessing and feature extraction processes needed in traditional machine learning approaches, which proves to be a great advantage for this task due to the several linguistic variations such as misspellings, multiple names, transliterations and missing titles that appear in the particular entity linking task.

Our approach lags in the cases with semantic proximity despite the ability of LMs to recognize semantically similar terms and sentence. This option can be problematic because it might lead to high scores for rejected candidate pairs. Another drawback of this approach is the fact that it ignores the linking of candidate entity pairs from a different language.

There are some future directions we would like to pursue next. As future work, our will is to extend the base dataset with even more candidate pairs. We plan to analyse further the mismatches to understand the limitations of the model or hidden domain knowledge that is omitted from the training process. Also, a nice contribution and extension of our approach is the introduction of language detection and translation to handle the multilingual entity linking. In addition, we plan to explore further the use of Named-Entity Recognition models, creating our own model, that will be tailor-made for the needs of the music industry.

REFERENCES

Bilenko, M. and Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48.

Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., and Stefanidis, K. (2019). End-to-end entity resolution for big data: A survey. *arXiv preprint arXiv:1905.06397*.

Cohen, W. W. and Richman, J. (2002). Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 475–480.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., and Tang, N. (2018). Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment*, 11(11):1454–1467.

Fu, C., Han, X., Sun, L., Chen, B., Zhang, W., Wu, S., and Kong, H. (2019). End-to-end multi-perspective matching for entity resolution. In *IJCAI*, pages 4961–4967.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hirschberg, J. and Manning, C. D. (2015). Advances in natural language processing. *Science*, 349(6245):261–266.

Jiang, N. and de Marneffe, M.-C. (2019). Evaluating bert for natural language inference: A case study on the commitmentbank. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6088–6093.

Konda, P., Das, S., Suganthan GC, P., Doan, A., Ardalan, A., Ballard, J. R., Li, H., Panahi, F., Zhang, H., Naughton, J., et al. (2016). Magellan: Toward building entity matching management systems. *Proceedings of the VLDB Endowment*, 9(12):1197–1208.

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Li, Y., Li, J., Suhara, Y., Doan, A., and Tan, W.-C. (2020). Deep entity matching with pre-trained language models. *arXiv preprint arXiv:2004.00584*.

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., and Raghavendra, V. (2018). Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, pages 19–34.

Panda, A. and Patel, A. (2012). Role of collective management organizations for protection of performers' right in music industry: In the era of digitalization. *The Journal of World Intellectual Property*, 15(2):155–170.

Papadakis, G., Papastefanatos, G., Palpanas, T., and Koubarakis, M. (2016). Boosting the efficiency of large-scale entity resolution with enhanced metablocking. *Big Data Research*, 6:43–63.

Primpeli, A., Peeters, R., and Bizer, C. (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.

Qu, C., Yang, L., Qiu, M., Croft, W. B., Zhang, Y., and Iyyer, M. (2019). Bert with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1133–1136.

Robertson, S. and Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sarawagi, S. and Bhamidipaty, A. (2002). Interactive deduplication using active learning. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–278.

Singla, P. and Domingos, P. (2006). Entity resolution with markov logic. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 572–582. IEEE.

Skog, D., Wimelius, H., and Sandberg, J. (2018). Digital service platform evolution: how spotify leveraged boundary resources to become a global leader in music streaming. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.