Analyzing How BERT Performs Entity Matching

Matteo Paganelli
University of Modena and Reggio Emilia
Modena, Italy
matteo.paganelli@unimore.it

Francesco Del Buono
University of Modena and Reggio Emilia
Modena, Italy
francesco.delbuono@unimore.it

Andrea Baraldi
University of Modena and Reggio Emilia
Modena, Italy
andrea.baraldi96@unimore.it

Francesco Guerra
University of Modena and Reggio Emilia
Modena, Italy
francesco.guerra@unimore.it

ABSTRACT

State-of-the-art Entity Matching (EM) approaches rely on transformer architectures, such as *BERT*, for generating highly contextualized embeddings of terms. The embeddings are then used to predict whether pairs of entity descriptions refer to the same real-world entity. BERT-based EM models demonstrated to be effective, but act as black-boxes for the users, who have limited insight into the motivations behind their decisions.

In this paper, we perform a multi-facet analysis of the components of pre-trained and fine-tuned BERT architectures applied to an EM task. The main findings resulting from our extensive experimental evaluation are (1) the fine-tuning process applied to the EM task mainly modifies the last layers of the BERT components, but in a different way on tokens belonging to descriptions of matching / non-matching entities; (2) the special structure of the EM datasets, where records are pairs of entity descriptions is recognized by BERT; (3) the pair-wise semantic similarity of tokens is not a key knowledge exploited by BERT-based EM models.

PVLDB Reference Format:
Matteo Paganelli, Francesco Del Buono, Andrea Baraldi, and Francesco Guerra. Analyzing How BERT Performs Entity Matching. PVLDB, 15(8): 1726 - 1738, 2022.
doi:10.14778/3529337.3529356

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/softlab-unimore/bert-attention-for-em.

1 INTRODUCTION

The adoption of BERT [7] and other transformer architectures [32] has resulted in a breakthrough in the effectiveness of the Entity Matching (EM) approaches (see, for example, [4, 17]). Nevertheless, BERT, and more generally transformers, are black-box architectures and it is not easy to understand which are the internal mechanisms that allow them to obtain such outstanding results. Providing an answer to this question is crucial to increase their trustworthiness and promote their application in real-world scenarios.

Proceedings of the VLDB Endowment, Vol. 15, No. 8 ISSN 2150-8097.
doi:10.14778/3529337.3529356

The NLP research community recently put a big effort in analyzing which knowledge is learned and applied by transformer-based architectures. The term BERTology [25] was coined to refer to the large number of papers that have investigated BERT-based architectures [6, 9, 13, 14, 16, 18, 19, 24, 28, 29]. These analyses typically follow two main research directions: they directly evaluate the contribution of specific architecture components (such as contextualized embedding or attention modules) or they examine the parameters of probing classifiers trained on top of them.

Inspired by these works, we propose to inspect the ability of BERT-based approaches to perform EM. This operation is usually conceived as a binary classification problem, where the class shows if pairs of entity descriptions are (or not) matching, i.e., they refer to the same real-world entity. The structure of the EM datasets, describing two evidences per record, makes this task far from the ones typically studied in ML and DL, whereas the records usually refer to single evidences. We wonder in which way the transformers are able to manage this special dataset structure and if the fine-tuning allows transformer components to learn some matching logic from the data. We therefore formulated three research questions that provide methodological guidance to our research. They concern (1) the impact of fine-tuning on the effectiveness of the EM task; (2) the capability of BERT to detect and exploit the special structure of the EM datasets and (3) the extent to which BERT-based EM models rely on the semantic similarity of pairs of tokens.

Sections 4-6 propose deep experimental evaluations that provide answers to the aforementioned questions. Their overall analysis allows us to get three main findings (Section 7): 1) BERT architectures fine-tuned on the EM task are more efficient than ad-hoc EM models. Fine-tuning impacts the last layers of the attention modules and modifies the space of the embeddings differently depending on whether the records refer to matching/non-matching entity descriptions. 2) The attention weights recognize that records in the EM datasets describe pairs of entities through the same set of attributes. A special pattern is found in the attention modules that gives attention to attributes describing the same entity property in different entity descriptions. Moreover, BERT is able to discover the attributes which mostly contribute to solving the EM task. 3) The pair-wise semantic similarity of tokens is not particularly exploited by the model. BERT seems to introduce and use a more contextualized, pragmatic kind of knowledge that involves more tokens and attributes. The importance of the one-to-one relationships defined

by the semantic similarity decreases with the fine-tuning process, and increases the importance given to the EM structure.

The rest of the paper is organized as follows. Section 2 introduces related work; in Section 3 we define the boundaries of our experimental evaluation that results in an analysis of the impact of fine-tuning on the effectiveness (Section 4); of what BERT learns from the dataset structure (Section 5); and the importance of the pair-wise semantic similarity of tokens in the EM process (Section 6). In Section 7 we point out some lessons learned, and in Section 8 we sketch out some conclusions and future work.

2 RELATED WORK

BERT architecture overview. BERT [7] is a large transformer-based architecture whose main component is a self-attention module [32]. It takes as input a sequence of token representations, learns the reciprocal attention that each token of the sequence directs towards each other token (i.e. the attention weights), and outputs a new sequence obtained from the weighted average between the original token representations and the attention weights. BERT organizes self-attention modules on multiple layers, and each of them is divided into multiple parallel "heads" which act on separate linear transformations applied to the same input sequence of token representations. Several variants of this architecture have been proposed where a different dimension (e.g., different number of layers, etc.) or a different vocabulary (i.e. cased or uncased) are used. In this paper we will refer to the bert-base-uncased variant, which consists of 12 layers, each having a hidden size of 768 and 12 attention heads (110M parameters). BERT is pre-trained on 3.3 billion tokens of English text to perform two tasks: the masking language modeling, which consists of predicting the token that has been masked by the input text, and the next sentence prediction, which consists of predicting the next sentence of an input text. Although this architecture can be used in a pre-trained form to obtain advanced token input representations, a fine-tuning process is usually applied. It typically consists of adding one or more fully connected layers to the BERT architecture and training the resulting network with respect to a reference task. In this paper, we will consider both the pre-trained version and a fine-tuned version created to solve an Entity Matching task.

BERT inspection. Recently, a thriving collection of works, identifiable under the term BERTology [25], has inspected the BERT architecture to assess its ability to learn correct linguistic artifacts. A first category of these works exploits probing classifiers built on top of different BERT intermediate representations (such as contextual embeddings or attention heads) to understand if these components capture specific linguistic patterns (e.g., dependencies between part-of-speech) [9, 19, 24, 28, 29]. From these studies, it emerged that BERT is able to encode a great variety of syntactic and semantic relationships in different regions of the network and in a hierarchical way (i.e. through syntactic tree structures) [13, 18]: simple syntactic information is captured in the first layers, while more complex relationships in the deeper layers. Despite these findings, the reliability of these probing tasks has recently been debated as their results can be easily misinterpreted [29] or perturbed by the evaluation methodology itself [5, 12, 19, 34]. Parallel to these works, several approaches directly inspected the BERT architecture [6, 14, 16]. Unlike probing models, they do not depend on any auxiliary supervised task and therefore they do not require additional training. The study proposed in this paper belongs to the latter category, and focuses on the analysis of BERT's data structures when employed to solve EM tasks. Although many studies analyze BERT's behavior in various tasks [10, 14], to the best of our knowledge, this is the first work about Entity Matching.

Deep Entity Matching. Deep Learning (DL) models can effectively address EM. DeepER [8] and DeepMatcher [21] were the pioneers of this kind of architecture. They leverage recurrent neural networks, possibly integrated with attention modules, to encode pairs of entities in multi-dimensional vectors and create a binary classifier based on the similarity of these embeddings to generate the matching decision. With the successful application of transformer architectures [32] in the NLP domain, EM models have also integrated this new technology. These are complex neural networks trained on large generalist corpus in a self-supervised manner, which are typically re-used in downstream tasks after the application of a fine-tuning process. Their application to EM tasks has pushed the state-of-the-art performance [22]. Some examples of BERT-based EM systems are [4, 17, 23]. In [4] the most recent transformer-based models are fine-tuned on the EM task, empirically demonstrating their high efficacy in solving the task even in dirty or textual datasets and without the need for a task-specific architecture. [17] proposes Ditto, which is now the most performing EM model proposed in the literature. It consists of a BERT architecture fine-tuned on the EM task which is further optimized by injecting domain knowledge (separators are added to mark the attributes in the EM entries), applying text summarization methods based on TF-IDF, and adapting data augmentation techniques for text to add (difficult) examples in the training data. In [23] a dual-objective training technique for BERT is proposed, which forces the model to predict the entity identifier in addition to the match / non-match decision. Recently these architectures have also found application in the blocking phase [30] and a survey on the adoption of DL architectures in EM is available in [1].

3 THE EXPERIMENTAL ANALYSIS

Methodology. We believe that answering the following three research questions can lead to understanding how BERT-based models support EM, what knowledge is learned through the tuning process, and how this knowledge improves the matching process.

(1) To what extent and for what reasons fine-tuning is able to improve the effectiveness of the results achieved by BERT-based EM models? (Section 4)
(2) Does BERT detect and exploit through the fine-tuning the specific structure of the EM datasets composed of pairs of entity descriptions, sharing the same set of attributes? (Section 5)
(3) How much does BERT rely on pair-wise semantic similarity of tokens, how this knowledge changes with the fine-tuning process? To what extent does this similarity support the EM process? (Section 6)

Table 1: Magellan Benchmark

Dataset	Type	Datasets	Size	% Match	Sample Size	# Attr
S-FZ	Structured	Fodors-Zagats	946	11.63	132	6
S-DG		DBLP-GoogleScholar	28,707	18.63	6414	4
S-DA		DBLP-ACM	12,363	17.96	2664	4
S-AG		Amazon-Google	11,460	10.18	1398	3
S-WA		Walmart-Amazon	10,242	9.39	1152	5
S-BR		BeerAdvo-RateBeer	450	15.11	80	4
S-IA		iTunes-Amazon	539	24.49	156	8
T-AB	Textual	Abt-Buy	9,575	10.74	1232	3
D-IA	Dirty	iTunes-Amazon	539	24.49	156	8
D-DA		DBLP-ACM	12,363	17.96	2664	4
D-DG		DBLP-GoogleScholar	28,707	18.63	6414	4
D-WA		Walmart-Amazon	10,242	9.39	1152	5

Datasets. We performed the experiments against the datasets provided by the Magellan library[1] which is the reference benchmark for the evaluation of EM tasks. The datasets describe pairs of entity descriptions sharing a common structure. We summarize in Table 1 some statistical measures describing the datasets, reporting for each of them the total number of records (fourth column) and the percentage of records associated to a match label (fifth column). In the experiments that require to train and evaluate the effectiveness of BERT in performing EM, we divided the datasets into train, validation and test sets with a proportion of 60, 20, 20. The remaining experiments, which apply a-posteriori analyses of BERT components, are instead performed on random samples of records, with a size depending on the dataset as reported in column *Sample Size* of Table 1. The samples are balanced, including the same number of matching and non-matching entity descriptions. The experiments were all repeated three times and the average value is reported in the paper.

Dimensions of the Analysis. The experimental evaluations are performed along 3 main dimensions, (1) data encoding, (2) data unit representation, and (3) model application. We tested two techniques for *encoding the data. Sentence-pair (SP)* consists of supplying BERT with two distinct phrases (separated by the special token [SEP]), where each phrase corresponds to the textual representation of an entity description obtained by concatenating all attribute values. The second, *attribute-pair (AP)*, modifies the previous approach by using the special token [SEP] to delimit the content of the attributes. In this way, we make BERT aware of the subdivision of information by attributes existing in the datasets. A similar encoding has also been adopted in [17]. We performed the experiments with different *granularities for the data representation.* In some tests, we evaluated the attention given to *tokens (TK).* In other tests, we aggregated the scores by the attribute they belong to. We experimented with two techniques for *representing the attention for attributes*: by considering the *average (AV)* of the attention given to their composing tokens or the *maximum value (MA).* Finally, concerning the *model*, we performed the experiments with both a *pre-trained (PT)* and a *fine-tuned (FT) BERT model.* The architecture of the pre-trained model is composed of two fully connected layers with 100 and 2 neurons respectively (where the 2 output neurons represent the match and non-match classes) added on the top of the original BERT's language model[2]. These additional layers have been trained on the EM task to predict whether pairs of input entities are matching, by keeping unaltered the BERT's original pre-trained model. The fine-tuned architecture consists of a single classification layer inserted on top of the embedding corresponding to the [CLS] token. This is the usual standard practice adopted for fine-tuning BERT to a downstream classification task [4, 17]. The whole architecture is here trained on the EM task, thus modifying the weights of the attention modules and the consequent embeddings of the original BERT model. An *experiment setting* is a proper selection of the dimensions of the analysis, namely $setting = (DE, AR, MO)$, where DE is one of the techniques implemented for data encoding (SP or AP), AR for the attribute representation (AV or MA) and MO for the model application (PT or FT).

Data Structures. The analysis of the attention modules relies on two special data structures that show the attention provided by the BERT-based architecture. The *attention head* [7] is a squared matrix with cells showing the attention scores that tokens in the rows give to the token in the columns. The BERT architecture consists of 12 layers each of which contains 12 heads, for a total of 144 attention heads. In the *attribute attention head* the attention scores are aggregated by attribute according to one of the techniques introduced (AV or MA). Since the EM dataset describes pairs of entities, attention matrices can be decomposed into four quadrants (see for example Figure 4). The top-left quadrant shows the attention given to the attributes of the first entity from the attributes of the same entity. The bottom-right quadrant describes the same information for the second entity. The bottom-left quadrant shows the attention given to the attributes of the first entity by the ones of the second entity and the top-right quadrant the opposite score: the attention to the first entity from the second one. The attention and the attribute attention matrices can be aggregated per layer (by averaging the values in all the heads) and per dataset (by averaging the attention data structures across all the records of a dataset).

Limitations. The pre-trained and fine-tuned EM models proposed are one of the simplest possible architectures based on the BERT model. This increases the generality and applicability of the findings obtained to all BERT-based architectures (e.g., *Ditto* [17] and [4]). Nevertheless, the analysis is affected by similar limitations as other works in the literature sharing the same methodology. Concerns have been raised about the methodology of inspecting individual components of such complex architectures. Studies have shown that the knowledge acquired by these models is spread throughout the entire architecture and an analysis of the individual components may not be sufficient [16]. In particular, the analysis of the role of attention modules in complex models has recently been discussed. [3, 15, 26] discovered that limited correlations exist between attention weights and the predictions of the model. This thesis is further exacerbated by the fact that in recent transformer architectures these modules are followed also by several non-linear

[1] https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md

[2] The application of the BERT model to a record generates a 768-dimensional embedding for each constituting token. A 768-dimensional vector is then generated for each entity in the description by averaging (across the last 4 layers) the embeddings of their associated tokens. A difference vector is then calculated by subtracting the representation of the first and second entity and supplied as input to the fully connected layer. The 768-dimensional vector is then compressed into a 100-dimensional representation and reduced via a softmax layer to a matching / non-matching probability score.

transformations. Nevertheless, this is a controversial point since other papers, as [31, 33], demonstrated that low correlations happen only in limited conditions.

4 IMPACT OF THE FINE-TUNING PROCESS ON THE EM TASK

The goal of this Section is to evaluate to what extent and in which way the fine-tuning process improves the ability of a BERT-based model to perform EM tasks. The first experiment proposed in Section 4.1 evaluates the effectiveness of pre-trained and fine-tuned BERT EM models by evaluating both the data encodings proposed for the entity descriptions. With the experiment in Section 4.2, we analyze the impact of fine-tuning on the attention modules by comparing the attention weights before and after the process. Finally, in Section 4.3, we evaluate if the fine-tuning impacts on the embeddings of matching and non-matching word pairs.

4.1 Effectiveness

Implementation. We evaluate BERT's ability to solve EM tasks by adding on top of its modules a binary classifier as described in Section 3. We experiment with 4 settings, obtained by varying the data encoding and the model application, i.e. $settings = (SP/AP, -, PT/FT)$. The results of the experiment are shown in Table 2, and compared with *DeepMatcher+* (*DM+*)[3] [21], a reference DL-based EM approach that does not rely on a transformer architecture, and *Ditto* [17], one of the best BERT- based EM approaches.

Discussion. Even if *DM+* obtains good results in most the datasets, the techniques based on BERT outperform them in particular in the resolution of textual and dirty datasets (i.e., the ones with a higher percentage of missing values and misalignment between attributes, identified with the prefix T and D in the Table). This result is consistent with the literature [4]. Moreover, *Ditto*, which extends BERT with data augmentation techniques and advanced EM data encoding, further improves the results. Finally, data encoding does not have on average a real impact on the results.

4.2 Attention

Implementation. To investigate the reasons that make the fine-tuned architecture so effective on the EM task, we now evaluate how the attention weights of a pre-trained BERT architecture change after the fine-tuning. To carry out the experiment we adapt the methodological procedure applied in [16] to perform NLP tasks. The cosine similarity between (flattened versions of) the attention heads associated to the pre-trained and fine-tuned models is computed for every head and layer of each dataset record. The average of these similarities for all the records in the dataset is shown in Figure 1 for the $settings = (SP, -, PT/FT)$.

Discussion. The heads that undergo the greatest variations are those located in the last layers (i.e. the overall similarity of the last layers is generally closer to 0). This is particularly evident for the structured and dirty versions of DG, DA and WA. The result suggests that more EM-specific information is encoded in the last layers, while shallow layers capture more general linguistic information mainly deriving from the pre-train. This finding is consistent with similar experiments performed in other NLP scenarios [10, 11, 16, 25].

[3] In Table 2, we consider the results published in [17].

Table 2: The effectiveness of pre-trained and fine-tuned BERT-based models: $settings = (SP/AP, -, PT/FT)$ **(F1 score).**

	Pre-trained (attr-pair)	Pre-trained (sent-pair)	Fine-tuned (attr-pair)	Fine-tuned (sent-pair)	DM+	Ditto
S-FZ	97.67	97.67	100.00	97.67	100.00	100.00
S-DG	92.80	92.40	94.92	94.78	94.70	95.80
S-DA	97.52	97.41	98.42	98.65	98.45	99.17
S-AG	65.19	63.26	70.21	68.52	70.70	75.58
S-WA	54.81	59.89	79.79	78.85	73.60	86.76
S-BR	82.76	82.76	77.78	84.85	78.80	94.37
S-IA	86.21	85.19	90.00	93.10	91.20	97.80
T-AB	62.35	59.50	81.42	83.51	62.80	89.79
D-IA	70.59	84.21	94.74	94.74	79.40	95.65
D-DA	96.85	96.10	98.43	98.42	98.10	99.08
D-DG	91.63	92.27	95.07	94.77	93.80	95.75
D-WA	56.60	50.76	79.59	77.33	53.80	85.69

4.3 Embeddings

Implementation. We complement the previous experiment by analyzing how the fine-tuning alters the space of the pre-trained embeddings. We expect that the increased ability of the model to solve the EM task to be reflected in the distribution of the embeddings of the words. We hypothesize that the fine-tuned model increases the similarity of the embeddings of the words appearing in matching pairs and vice versa decreases the one of words occurring only in non-matching pairs. To analyze the validity of this consideration, we selected a sample of 1000 pairs of random words (where the first word is selected from the left entity and the second from the right one) that occur exclusively in matching, non-matching records. We also performed an analysis on random records to provide a baseline. We then calculated the (cosine) similarity of their embeddings and evaluated the percentage of times in which it is higher than 0.7 (threshold we choose to describe words with a medium-high similarity). The results of this experiment for the $settings = (SP, -, PT/FT)$ are shown in Figure 2.

Discussion. First of all, we observe how the fine-tuning process increases the similarity between all pairs of tokens examined compared to the pre-trained version. Secondly, we notice how the similarity between the pairs of tokens belonging to records describing matching entities is on average higher than those of the tokens occurring in non-matching and random records. At the same time, the similarity associated with non-matching pairs is lower than the other categories of records. This is because the number of similar words in descriptions of non-matching entities is generally less than in matching entities. Despite this, we observe that there is a high variability in the results: there are pairs of tokens with a high similarity regardless the fact they belong to descriptions of matching or non-matching entities.

5 RELYING ON THE DATASET STRUCTURE

The experiments in this Section evaluate if the fine-tuning process detects and exploits the special data structure adopted by the EM datasets. In particular, with the experiment in Section 5.1 we investigate whether and to what extent the fine-tuned BERT model

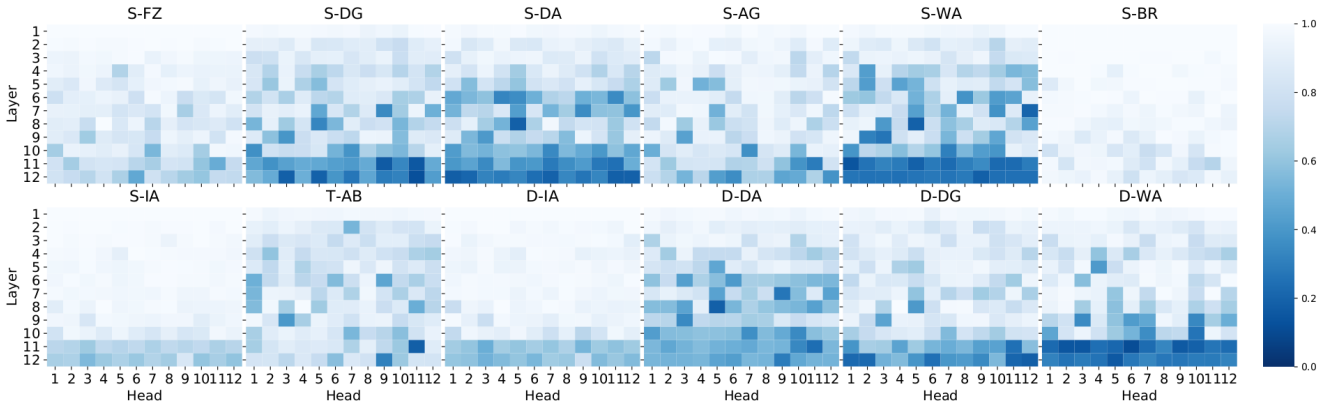


Figure 1: Similarity between pre-trained and fine-tuned attention scores, $settings = (SP, -, PT/FT)$. The darker the cell, the greater the difference between the attention scores of fine-tuned and pre-trained models.

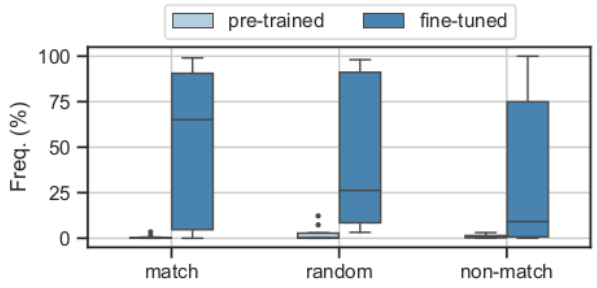


Figure 2: Impact of the fine-tuning on the similarity of the embeddings: $settings = (SP, -, PT/FT)$. The y-axis shows frequency of the similar embeddings in the entity descriptions.

exploits the relationships between the pairs of entities that appear in each EM data entry. We therefore analyze the attention given to the pairs of tokens belonging to two different entity descriptions in the same EM record and we evaluate the changes determined by the fine-tuning. The experiments described in Section 5.2 study the presence of frequently occurring patterns in the BERT's attention modules. In particular, we analyze the patterns that show relationships between attributes. The experiments in Section 5.3 evaluate wheater the attention provided by the pre-trained and fine-tuned BERT models reflects a different contribution of the attributes in performing the EM task.

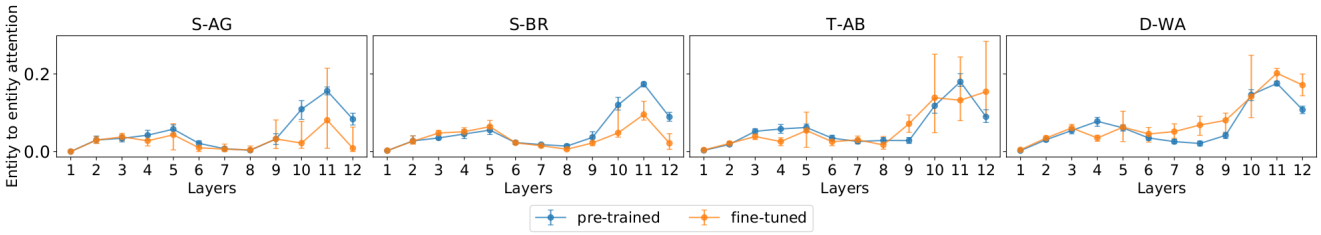
5.1 The entity-to-entity (E2E) pattern

Implementation. The goal is to discover if there is attention between the pairs of entities described in the same record. We expect this to be a frequent pattern in EM datasets where the task is to identify the correspondences between the tokens of two entities belonging to the record. The idea is to understand: 1) the contribution of this pattern in the attention generated by BERT, and 2) which are the layers where the pattern is mainly active. To perform the experiment, we build an average attention head for each layer by averaging the scores of all its heads. Then, for each average attention head, we count the percentage of cells referring to tokens from the descriptions of different entities and having an attention score in the last quintile of the matrix. The scores are then averaged considering the dataset records.

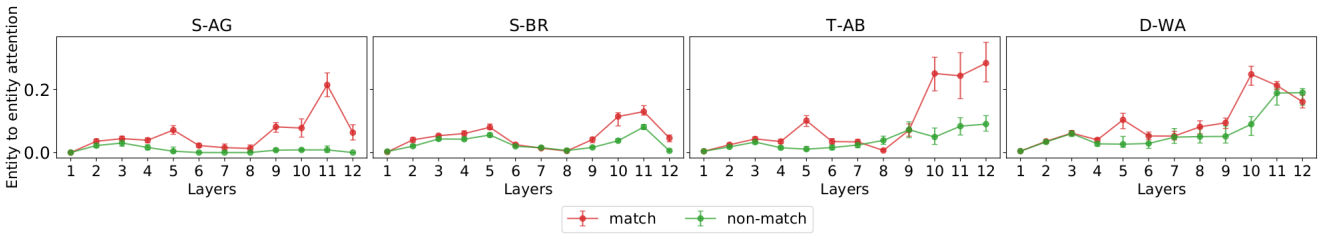
Discussion. Figure 3a shows the entity-to-entity attention pattern generated by the pre-trained and fine-tuned models on a selection of the datasets. First of all, we observe how the entity-to-entity attention pattern contributes to a maximum of 30% on the entire attention produced by BERT. Then, this pattern mainly occurs in the last 3 layers of the architecture, suggesting that BERT uses these layers to encode "cross-entity" information. This trend is confirmed by both the pre-trained and fine-tuned models, which generate very similar absolute values, suggesting that this behavior is inherited almost exclusively from the initial training of the architecture. The impact of the fine-tuning on the pattern largely depends on the dataset. In some cases, the entity-to-entity attention pattern is more marked in fine-tuned models, in other cases on the pre-trained. Nevertheless, we observe that the interquartile range markedly increases with the layer in almost all datasets and especially for the fine-tuned models. In Figure 3b we inspect the variability introduced by the fine-tuning by comparing the intensity of the pattern for records referring to matching and non-matching entities. The diagrams show that the high variability is due to a diversified contribution to the E2E pattern from matching/non-matching records. This therefore suggests that the fine-tuned model learned to distribute attention in a different way according to the type of record.

5.2 The attention patterns involving the dataset attributes

Implementation. The goal is to observe if there are frequently occurring patterns in the BERT's attention modules when applied to EM tasks. A special matrix is introduced with the aim of providing a compact and clear representation of the most expressive patterns for a dataset. The matrix is built upon the attribute attention heads, where the actual values are substituted by a boolean mask showing the pairs of attributes measuring an attention score above the average. There is a boolean mask for each record, head and layer. We



(a) Pre-trained vs fine-tuned models, $setting = (SP, -, PT/FT)$.



(b) Matching vs. non-matching entities, $setting = (SP, -, FT)$.

Figure 3: Entity-to-entity attention. The y-axis reports the normalized number of times where pairs of tokens from descriptions of different entities are assigned an attention score in the last quintile of an average attention head calculated on a certain layer.

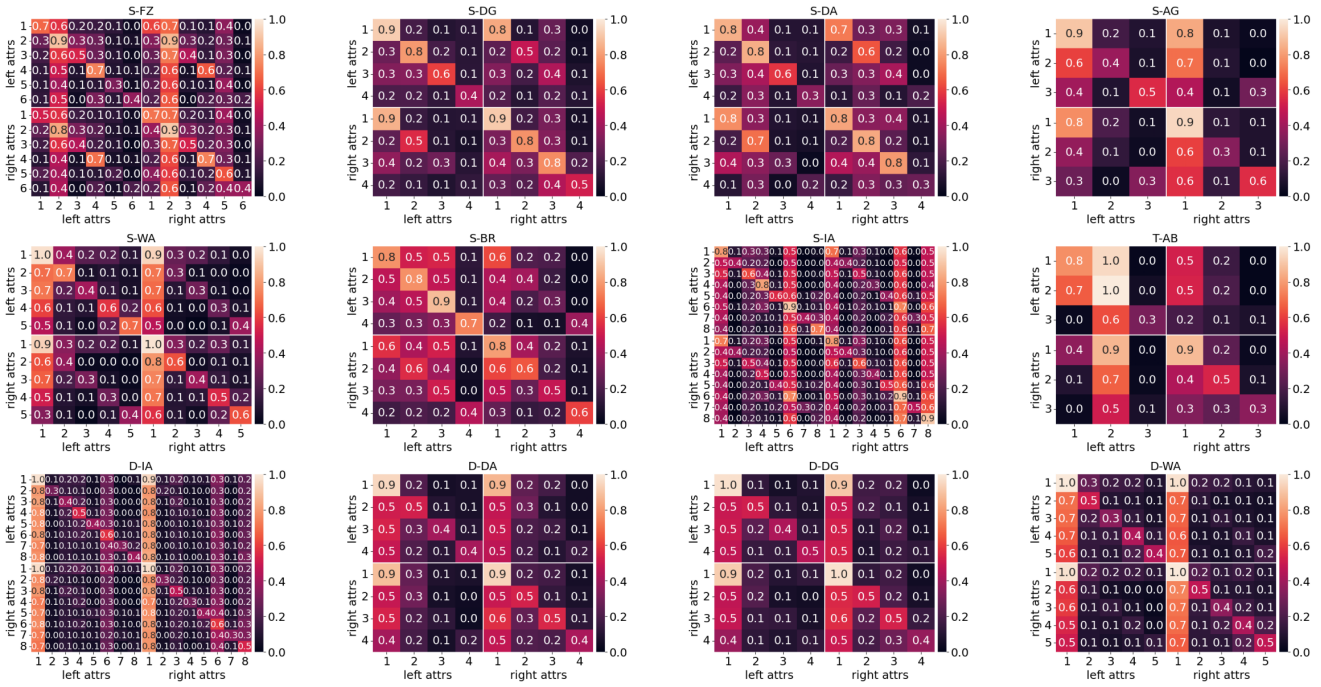


Figure 4: Attention between attributes: $setting = (SP, MA, PT)$. The cells show the attention given by the attribute on the row to the attribute on the column, divided by left and right entity. The lighter the color, the higher the attention.

average these masks over the 12x12 grid to generate a single mask for each record and then we average the masks across all records.

Discussion. Figure 4 shows the average boolean mask for all datasets with the $setting = (SP, MA, FT)$. Similar results are obtained with the other settings. The visual inspection of the matrices shows the existence of four main occurring patterns: (1) **diagonal**: high scores on the main diagonal (and close elements) of the matrices are obtained. This means that an attribute gives high attention towards itself and neighboring attributes. (2) **vertical**: vertical lines

show that all attributes have high attention towards the same target attribute. (3) **diagonal+vertical**: the diagonal and vertical patterns jointly occur in almost all datasets. (4) **matching attribute attention (MAA)**: there are elements with high scores in the main diagonals of the bottom-left and top-right quadrants composing each matrix. The first three patterns have been already observed as frequently occurring in other experiments concerning the analysis of NLP tasks [16]. The MAA is a new pattern emerging in the EM scenario: an attribute gives high attention towards its corresponding attribute (i.e., the matching attribute) of the other entity. This is because, by construction, the entity descriptions share a common schema and the matching attributes have the same relative positions in the dataset (i.e., dividing the attributes of an EM entry in two ordered set, one for each entity, matching attributes share the same position in both the sets).

To provide a measure of the consistency of the patterns in the datasets, we evaluated the frequency of the vertical, diagonal and matching patterns in all experimental settings. In particular, given a layer and head for a specific setting, we considered a vertical pattern as existing if there is a column having all values greater than the average of the attribute attention head; a diagonal pattern as existing if the average scores of the elements in the main diagonal are greater than the other diagonals; and, a MAA pattern if the average scores of the elements in the main diagonal of the top-right or bottom-left sub-matrices are greater than the other diagonals in the same sub-matrix. Figure 5 shows the percentage of the attribute attention heads where the patterns are found. The experiment shows that data encoding does not largely affect the results, and that the diagonal is the most common pattern. This is somewhat expected: this means that the terms give attention to themselves and to the other terms in the same attribute. The new MAA pattern is the second frequently occurring pattern in all datasets. This means that the structure of the EM dataset is recognized by BERT and preserved with the fine-tuning. Moreover, the dirty versions of the datasets show a reduced frequency of this pattern with respect to their structured versions, due to the misalignment of the values.

5.2.1 *The MAA pattern.*

We perform a deeper analysis of the MAA pattern, by analyzing its localization in the BERT layers and evaluating its contribution to the effectiveness of the model.

Localization. Figure 6 shows how the frequency of the MAA pattern varies across the layers of the architecture (the $setting = (SP, MA, PT/FT)$ is reported). We observe that the fine-tuning process leads to a reduction of the occurrences of the MAA pattern, in particular in the last three layers. This appears as a sort of counter-intuitive result: we would expect fine-tuning to introduce attention to the matching attributes. Nevertheless, with the experiment in the next section we show that this knowledge is crucial for the effectiveness of the model.

Impact of the MAA pattern on the effectiveness. Although previous experiments showed that the MAA pattern occurs less frequently in the fine-tuned model than in the pre-trained version, below we want to understand whether and to what extent it contributes to the capability of the model to perform the EM task. To carry out the experiment, firstly we calculate the average frequency of occurrence of the MAA pattern in the attribute attention heads. Then we remove a number of heads in descending order with respect to the pattern frequency, and we evaluate the variations of F1 score of the fine-tuned model. We compare these results with 2 baselines. The first (random) consists in pruning the same number of randomly selected heads. In the second baseline (importance) we prune the same number of heads, but according to their importance (as defined in [20]). In the experiments, we remove an increasing number of heads (5, 10, 20 and 50) and we evaluate the effectiveness of the model. Note that the pruning reduces the overall number of parameters of the BERT architecture from $108.5M$ to $99.6M$. The results of this experiment are reported in Figure 7 in the $setting = (SP, MA, FT)$.

We observe that the masking techniques generate a diversified impact on the performance of the model: while the random heads' removal does not determine substantial variations in the F1 score, the other techniques generate substantial reductions in the effectiveness of the model. This is particularly evident in the S-DG, S-DA, S-BR, T-AB, D-DA and D-WA datasets, where F1 scores close to zero are obtained. In these scenarios, despite some fluctuations, it is possible to observe how the removal of heads with the highest frequency of occurrence of the MAA pattern produces a more drastic reduction in performance compared to the two considered baselines. In many cases this behavior is noticeable even after removing only 5 heads. This result could provide a justification for the previous open problem: the fine-tuned model reduces the number of heads exhibiting the MAA pattern, but the information encoded within these heads is more largely used by the model to solve the EM task.

5.3 Importance of the attributes

This Section aims to investigate if BERT can recognize that not all the attributes have the same importance in performing the EM task.

5.3.1 *Analysis of the attention.*

Implementation. Through this experiment we evaluate the attributes of the dataset on which BERT relies when performing the EM task by inspecting the attention modules. To answer this question, we analyze the attention given by the special [CLS] token to the other tokens constituting the attributes of the EM dataset. The [CLS] token is a special token that is added by BERT to each input and is typically used to compute the prediction of any classification task. For this reason, the embedding associated to the [CLS] token is considered as a summary of the input sentence. The experiment considers only the last layer of the BERT architecture. We average the values on the attribute attention heads, and we select the attention values of the [CLS] token towards the attributes. We then compute the aggregated values by averaging the attention scores for all the dataset entries. We performed the evaluation with the $settings = (SP, MA, PT/FT)$ and in Figure 8a we report the results for a selection of the datasets.

Discussion. We observe that the pre-trained and fine-tuned models generate similar scores. This represents an unexpected result as the embeddings associated to the [CLS] tokens of fine-tuned models should be different from the ones of pre-trained models, since encoding task-specific information. The coarse-grained aggregation applied to attention scores that refer to the same attribute could be

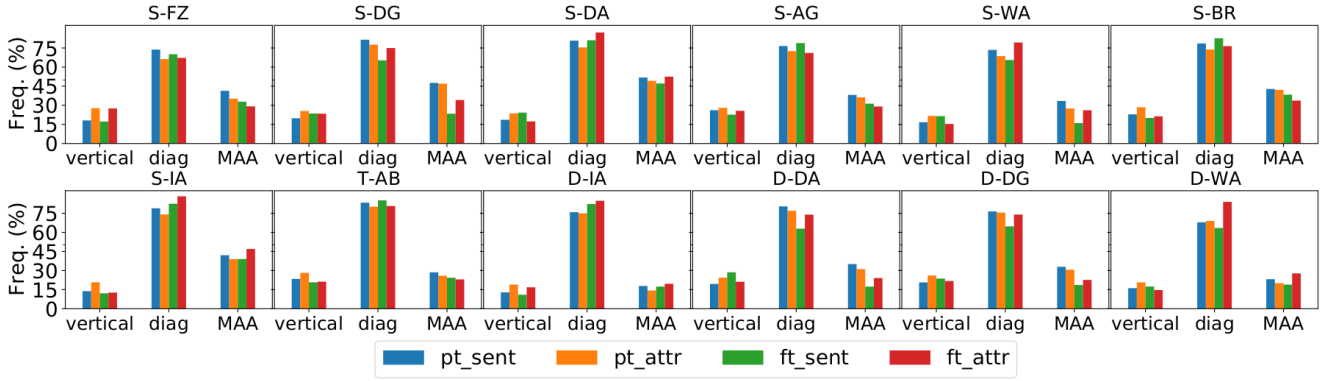


Figure 5: Comparison of pattern frequency in all the settings $(AP/SP, MA, PT/FT)$. The bars shows the percentage of the attribute attention heads where the patterns are found.

Figure 6: Frequency of the MAA pattern: $setting = (SP, MA, PT/FT)$. The diagrams show per layer the percentage of attribute attention heads where the pattern is found.

the reason for such a result. Nevertheless, the importance scores are consistently assigned to the attributes that, according to our domain knowledge, better allow users to identify matching entities. Moreover, in all datasets the attention generated towards the entity descriptions is symmetric (i.e., the attention is not focused on (attributes of) one of the two entities). Finally, we observe that the attention scores for the structured and dirty version of the DA dataset are diversified: on the dirty dataset the attention is exclusively towards one attribute; while on the structured version to multiple attributes. To elaborate on the analysis, Figure 8b shows the attention scores of the fine-tuned model differentiated between matching and non-matching entities. The scores are diversified and it is not possible to observe if there is more attention on records referring to matching/non-matching entities. However, we observe that in some cases the attributes receiving more attention change if we consider matching/non-matching entities. This is the case of the S-DG and S-DA datasets, where non-matching records give high importance to the attribute describing the authors of the publication, and matching records to the title.

5.3.2 *Gradient analysis.*

Implementation. The previous experiment showed how the attention scores associated with the attributes are consistent with the human evaluation. However, the experiment does not reveal whether the knowledge of the attribute importance is actually used in the inference of the matching decision. With this experiment we provide an answer to this question by analyzing the gradient of the attributes with respect to the predictions of the fine-tuned BERT

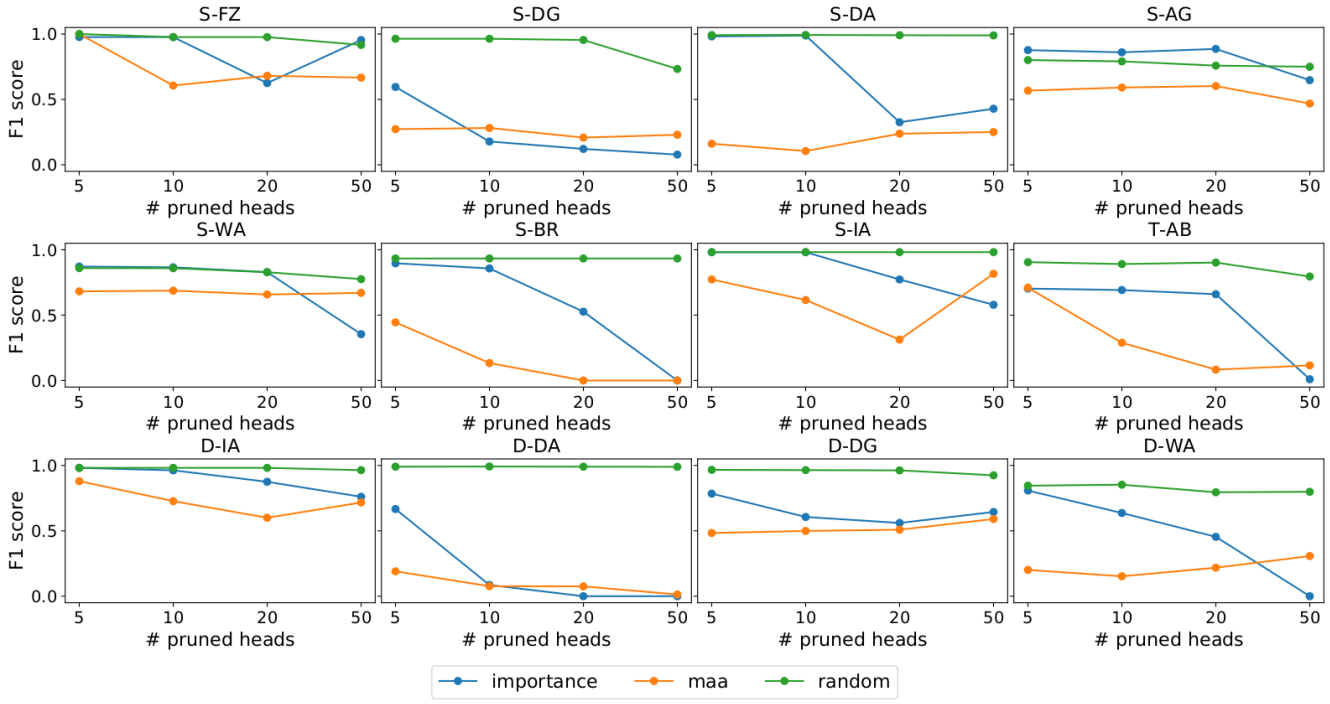
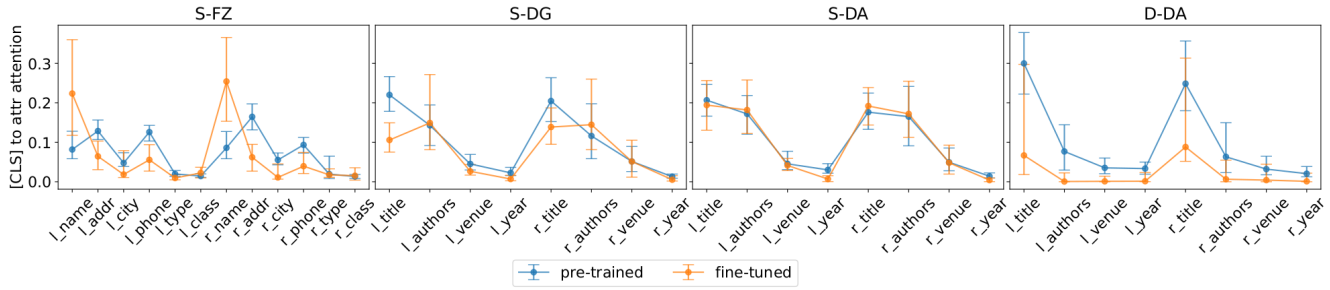
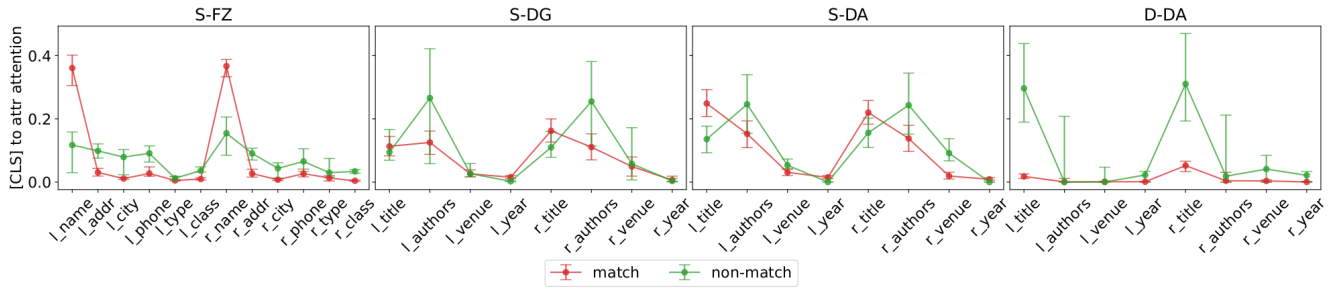


Figure 7: Impact of the MAA pattern on the effectiveness (F1-score) of the EM task performed with fine-tuned BERT models.



(a) Pre-trained vs fine-tuned models, $setting = (SP, MA, PT/FT)$.



(b) Matching vs. non-matching entities, $setting = (SP, MA, FT)$.

Figure 8: Attention given by the token [CLS] to the dataset attributes. The diagram shows the average value computed on all dataset entries of the attention registered in the last layer of the attention heads.

model. The gradient of a function output with respect to its input variable provides a measure of its contribution to the result. It represents the typical attribution method applied on neural architectures, which do not provide explicit features of importance

(such as the coefficients of a linear model), to determine the impact of single components on model predictions. In the experiment, we firstly select a balanced sample of 50 records for the matching and non-matching classes. Then, we compute the integrated gradient [27] of all tokens. We consider the gradient of each dataset attribute as the maximum gradient measured among its constituent tokens. The results of the experiment are shown in Figure 9.

Discussion. We observe that the results are consistent with those obtained in the previous experiment: the majority of the datasets rely on the tokens belonging to the first attribute to generate the prediction. We recall that by construction the first attribute of each dataset contains the most discriminative information for the entities (e.g., the attribute title is the first attribute in dataset S-DG).

6 EXPLOITATION OF THE SEMANTIC SIMILARITY KNOWLEDGE

The experiments in this Section allow us to understand if BERT introduces some semantic knowledge in the attention heads and embeddings to be used for identifying similar tokens thus supporting the EM task. For performing the experiments, we identify semantically similar pairs of terms by exploiting the cosine similarity of the token embeddings generated with the fastText model [2] and we analyze how BERT treats these inputs. In Section 6.1 we examine if BERT exploits semantic knowledge by evaluating the percentage of similar pairs found in the token pairs with the highest attention and how this amount changes with the fine-tuning. In Section 6.2 we analyze the correlation between the cosine similarity of the embeddings generated with the fastText model with the ones generated with BERT (pre-trained and fine-tuned). Finally, in Section 6.3 we perform a gradient analysis to evaluate the contribution of the semantic relationships on the inferences.

6.1 Attention and semantic similarity

Implementation. The goal of this experiment is to evaluate the extent of the attention that BERT gives to words with high similarity in solving the EM task. For each dataset record, we create two sets: one including the pairs of words with the highest attention score, i.e. the ones in the last quartile according to the values computed on an average attention head obtained by averaging all heads referring to the same layer; and the second with the most semantically similar pairs of words, i.e. the ones in the last quartile computed measuring the cosine similarity of their fastText embeddings. The experiment measures the percentage of shared pairs in the sets. Figures 10a and 10b show the results of the experiment, aggregated per dataset and per layer, respectively on the $setting = (SP, -, PT/FT)$.

Discussion. Figure 10a shows that generally the pre-trained models generate a percentage of shared pairs of words higher than the fine-tuned. Figure 10b shows that fine-tuning process largely decreases in the last layers the attention to pairs of highly similar tokens. The results of this experiment are somewhat unexpected since other experiments made on NLP tasks [13, 18] demonstrate that the semantic knowledge (1) is located in the last layers and (2) is largely exploited by transformers. We believe that BERT focuses on another kind of knowledge where the pragmatics complements the semantics and with a higher granularity than the one offered by the one-to-one token similarity. This assumption is confirmed by the fact that the fine-tuning improves the effectiveness of the results (see the experiments in Section 4.1) and that the deletion of the heads with the highest presence of the MAA pattern largely decreases the results (see the experiments in Section 5.2.1).

6.2 Embeddings and semantic similarity

Implementation. This experiment complements the one reported in the previous Section by analyzing the relationship between highly similar tokens, as resulting with the fastText embeddings, and the BERT embeddings. In particular, the goal is to evaluate if (1) semantically similar pairs of terms, according to fastText, give rise to close BERT embeddings and (2) if and how the fine-tuning changes the process. Note that this experiment differs from the one in Section 4.3, since it affects pairs of semantically similar terms instead of random tokens from matching and non-matching entity descriptions. The results of the experiment are shown in Figure 11 for the $settings = (SP, -, PT/FT)$, where, for sake of simplicity, we reported only the pairs with a cosine similarity greater than 0.7 according to the fastText encodings.

Discussion. The visual inspection of the distributions does not show any correlation between semantic similarity and the BERT embeddings. This result confirms the findings of the previous experiment: there is no correlation between the similarity of the BERT embeddings and the semantic similarity of the tokens. This happens even for tokens with the highest semantic similarity, which correspond in some cases to pairs of tokens with a similarity of the embeddings close to zero or negative. Finally, we observe how the fine-tuning process has significantly modified the embeddings space: in many datasets (with the exception of T-AB and D-WA and S-AG) the similarity of BERT's embeddings has grown considerably.

6.3 Gradient and semantic similarity

Implementation. The analysis of the gradient allows us to evaluate the actual contribution of the semantically similar pairs of tokens on the inferences performed by the EM classification model. As in the experiment in Section 5.3.2, we use the technique described in [27] to calculate the gradients associated with all the tokens belonging to the EM records. We then select exclusively the gradients associated with the pairs of words with a cosine similarity of the relative fastText embeddings greater than 0.7 and we sum these values to obtain a gradient for each pair of terms. In Figure 12 we compare the distribution of gradients with respect to the similarity of the fastText embeddings related to the pairs of words in the $setting$ $(SP, -, FT)$.

Discussion. The experiment shows that there is a low correlation between the semantic similarity between the tokens and a high value for the gradient. This confirms the findings of the previous experiments: the semantic similarity of the tokens is generally not taken into account by the BERT model and is not exploited for supporting the EM task.

7 LESSONS LEARNED

Summarizing the results obtained from the experiments, we observe that even if BERT-based architectures represent a breakthrough in performing EM (Section 4.1), the reasons why they largely support

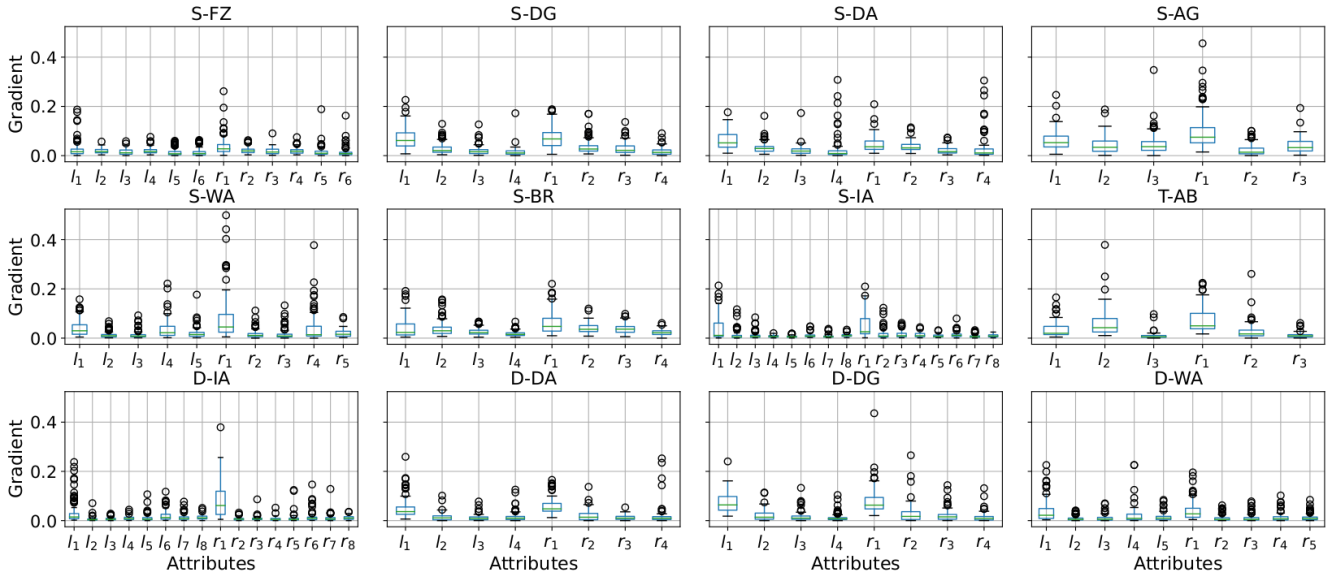
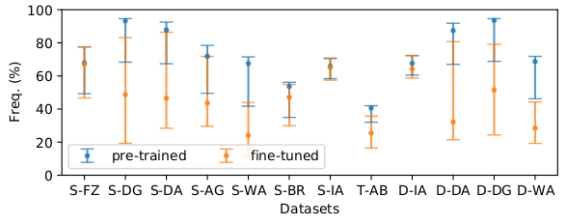
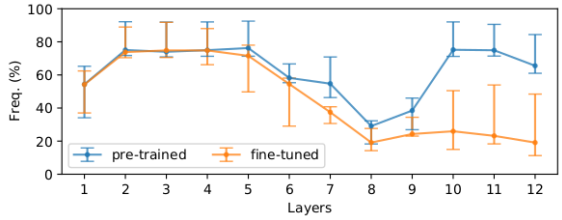


Figure 9: Importance given to the attribute computed with the gradient analysis. The highest the value, the highest the importance of the attribute for the prediction.



(a) Frequency across the datasets.



(b) Frequency across the layers.

Figure 10: Attention to word pairs with high semantic similarity. The y-axis shows the percentage of word pairs with the highest attention scores that are also highly semantically similar, according to their fastText embeddings.

the process can be only partially explained. Answering the three questions introduced in Section 3 allows us to observe:
(1) The fine-tuning process is crucial for improving the effectiveness of the BERT-based EM models. The experiments in Section 4.2 show that EM-specific knowledge is mainly encoded in the last layers of the architecture (as already observed in analyzing the BERT's behavior in performing other NLP tasks [10, 11, 16, 25]) and the embedding space changes with the fine-tuning. This leads to an increase in the number of semantically similar pairs of words found in different entity descriptions (Section 4.3).
(2) The specific structure of the EM datasets as composed of descriptions of pairs of matching / non-matching entities is recognized and exploited for performing the EM task. The experiments clearly show that not only the attention is given to tokens in the same EM record and belonging to different entity descriptions (Section 5.1) but also that matching attributes are recognized (Section 5.2). The analysis of the matching attribute attention pattern let emerge an unexpected result: a pattern, the MAA, identifying the matching attributes is found and despite its frequency in the datasets decreases with the fine-tuning, we observe that this knowledge represents a pillar for the effectiveness of the EM process (Section 5.2.1). Moreover, BERT can see that not all attributes are equally important in the EM process and that the importance of the attribute varies if we consider matching and non-matching entity descriptions (Section 5.3).
(3) The semantic similarity of the tokens is not a key knowledge for the EM process. This is another unexpected result: the attention to semantically similar tokens decreases with the fine-tuning (Section 6.1), and we did not find any correlation between the semantically similar embeddings computed with the fastText and the BERT approaches. Finally, the EM model does not rely on the pair-wise semantic similarity relationships between tokens (Section 6.3). The model seems to focus on a more contextualized kind of knowledge where pragmatic knowledge (discovered by BERT) complements the semantic knowledge.

8 CONCLUSION

In this paper, we analyzed the BERT's behavior in performing Entity Matching with the aim of understanding the reasons for its high performance. We discovered that BERT can recognize the structure

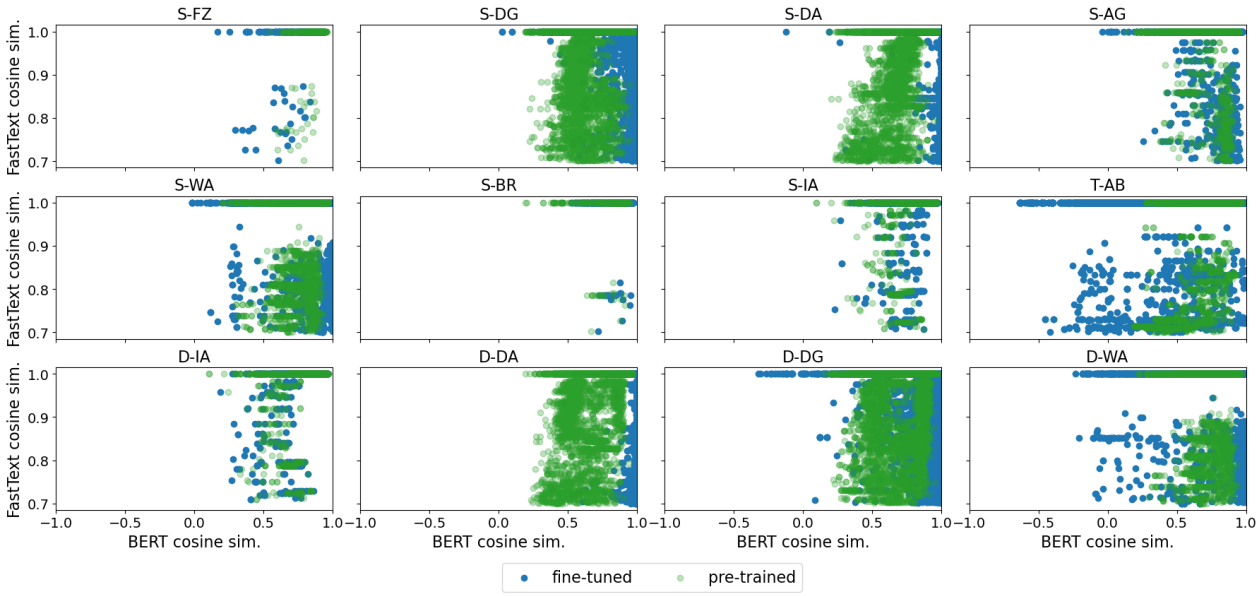


Figure 11: Comparison between BERT and fastText embeddings. The diagrams show the cosine similarity of the embeddings generated by BERT for word pairs with cosine similarity greater than 0.7 according to the fastText encodings.

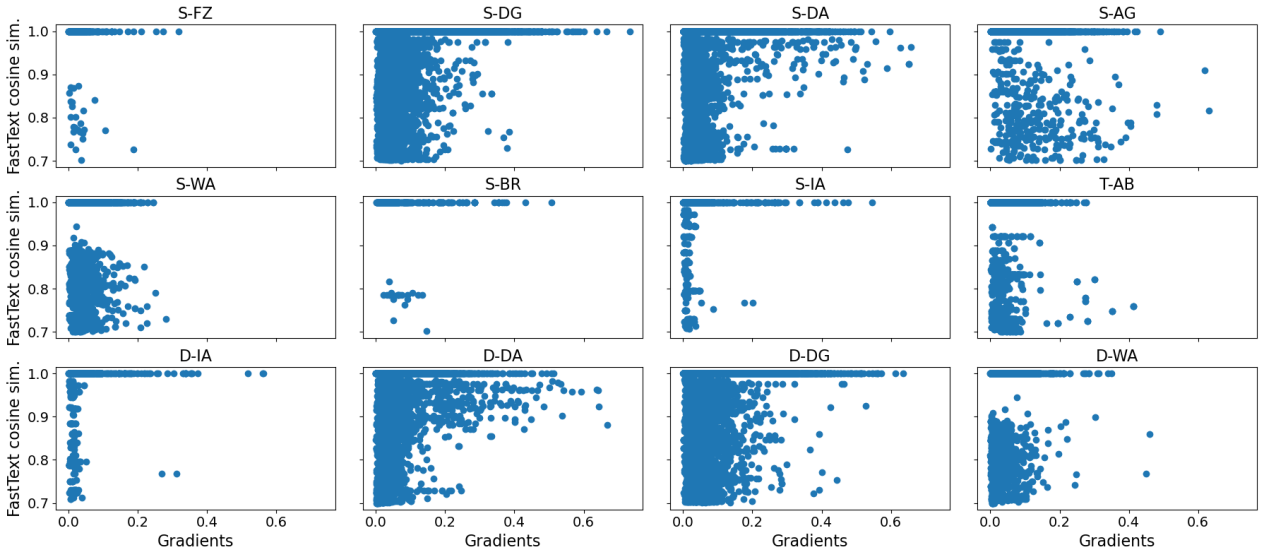


Figure 12: Comparison between gradient and semantic similarity. The gradients generated by a fine-tuned BERT model are compared with the cosine similarity of the fastText embeddings of pairs of words (only similarities greater than 0.7 are shown).

of EM datasets and extracts from the entity descriptions semantic knowledge that goes beyond the pair-wise association between tokens. Moreover, through the fine-tuning process, BERT learned to distribute the attention depending on whether the descriptions refer to matching or non-matching entities.

Future work will be devoted to further clarifying the reasons that make this architecture so performing on the EM task by 1) identifying the components that contribute most to the matching predictions, 2) experimenting with different fine-tuning approaches and 3) correlating the attention weights with the effectiveness of the prediction.

ACKNOWLEDGMENTS

This work was partially funded by the Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia (project AWESOME - FAR2021).

REFERENCES

[1] Nils Barlaug and Jon Atle Gulla. 2021. Neural Networks for Entity Matching: A Survey. *ACM Trans. Knowl. Discov. Data* 15, 3 (2021), 52:1–52:37.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics* 5 (2017), 135–146.

[3] Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On Identifiability in Transformers. In *ICLR*. OpenReview.net.

[4] Ursin Brunner and Kurt Stockinger. 2020. Entity Matching with Transformer Architectures - A Step Forward in Data Integration. In *EDBT*. OpenProceedings.org, 463–473.

[5] Steven Cao, Victor Sanh, and Alexander M. Rush. 2021. Low-Complexity Probing via Finding Subnetworks. *CoRR* abs/2104.03514 (2021).

[6] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. *CoRR* abs/1906.04341 (2019).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[8] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. *Proc. VLDB Endow.* 11, 11 (2018), 1454–1467.

[9] Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *CoRR* abs/1901.05287 (2019).

[10] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. Visualizing and Understanding the Effectiveness of BERT. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 4141–4150.

[11] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Investigating Learning Dynamics of BERT Fine-Tuning. In *AACL/IJCNLP*. Association for Computational Linguistics, 87–92.

[12] John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 2733–2743.

[13] John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4129–4138.

[14] Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do Attention Heads in BERT Track Syntactic Dependencies? *CoRR* abs/1911.12246 (2019).

[15] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. *CoRR* abs/1902.10186 (2019).

[16] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 4364–4373.

[17] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *Proc. VLDB Endow.* 14, 1 (2020), 50–60.

[18] Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting Inside BERT's Linguistic Knowledge. *CoRR* abs/1906.01698 (2019).

[19] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In *NAACL-HLT (1)*. Association for Computational Linguistics, 1073–1094.

[20] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One?. In *NeurIPS*. 14014–14024.

[21] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *SIGMOD Conference*. ACM, 19–34.

[22] Matteo Paganelli, Francesco Del Buono, Marco Pevarello, Francesco Guerra, and Maurizio Vincini. 2021. Automated Machine Learning for Entity Matching Tasks. In *EDBT*. OpenProceedings.org, 325–330.

[23] Ralph Peeters and Christian Bizer. 2021. Dual-Objective Fine-Tuning of BERT for Entity Matching. *Proc. VLDB Endow.* 14, 10 (2021), 1913–1921.

[24] Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting Contextual Word Embeddings: Architecture and Representation. In *EMNLP*. Association for Computational Linguistics, 1499–1509.

[25] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Trans. Assoc. Comput. Linguistics* 8 (2020), 842–866.

[26] Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable?. In *ACL (1)*. Association for Computational Linguistics, 2931–2951.

[27] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *ICML (Proceedings of Machine Learning Research)*, Vol. 70. PMLR, 3319–3328.

[28] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *ACL (1)*. Association for Computational Linguistics, 4593–4601.

[29] Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *ICLR (Poster)*. OpenReview.net.

[30] Saravanan Thirumuruganathan, Han Li, Nan Tang, Mourad Ouzzani, Yash Govind, Derek Paulsen, Glenn M. Fung, and AnHai Doan. 2021. Deep Learning for Blocking in Entity Matching: A Design Space Exploration. *Proc. VLDB Endow.* 14, 11 (2021), 2459–2472.

[31] Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention Interpretability Across NLP Tasks. *CoRR* abs/1909.11218 (2019).

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[33] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 11–20.

[34] Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed Masking: Parameter-free Probing for Analyzing and Interpreting BERT. In *ACL*. Association for Computational Linguistics, 4166–4176.