



A multi-facet analysis of BERT-based entity matching models

Matteo Paganelli¹ · Donato Tiano² · Francesco Guerra²

Received: 31 January 2023 / Revised: 17 August 2023 / Accepted: 29 October 2023 / Published online: 29 November 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

State-of-the-art Entity Matching 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; (4) fine-tuning SBERT, a pre-trained version of BERT on the sentence similarity task, i.e., a task close to EM, does not allow the model to largely improve the effectiveness and to learn different forms of knowledge. Approaches customized for EM, such as Ditto and SupCon, seem to rely on the same knowledge as the other transformer-based models. Only the contrastive learning training allows SupCon to learn different knowledge from matching and non-matching entity descriptions; (5) the fine-tuning process based on a binary classifier does not allow the model to learn key distinctive features of the entity descriptions.

Keywords BERT · Entity matching · Data integration · Transformers

1 Introduction

The adoption of BERT [11] and other transformer architectures [48] has resulted in a breakthrough in the effectiveness of the Entity Matching (EM) approaches (see, for example, [5, 24]). 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.

The NLP research community recently put a big effort in analyzing which knowledge is learned and applied by transformer-based architectures. The term BERTology [40] was coined to refer to the large number of papers that have investigated BERT-based architectures [9, 10, 14, 18, 19, 23, 25, 26, 37, 43, 44, 51]. These analyses typically follow two main research directions: they adopt an experimental methodology to 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 [40].

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

✉ Francesco Guerra
francesco.guerra@unimore.it

Matteo Paganelli
matteo.paganelli@hpi.de

Donato Tiano
donato.tiano@unimore.it

¹ Hasso Plattner Institute, University of Potsdam, Potsdam, Germany

² University of Modena and Reggio Emilia, Modena, Italy

allows transformer components to learn some matching logic from the data. In [34], we addressed the problem by analyzing how BERT performs the EM task according to three perspectives. This paper extends the analysis by adding two other research perspectives. The result is five 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; (3) the extent to which BERT-based EM models rely on the semantic similarity of pairs of tokens, (4) the impact of the technique used to pre-train the model on the ability of the transformer to learn the EM task and (5) which fine-tuning technique allows the model to better learn that its inputs are pairs of entity descriptions.

Sections 4–8 propose deep experimental evaluations that provide answers to the aforementioned questions. Their overall analysis allows us to get five main findings (Sect. 9): (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. (4) The effectiveness of the BERT-based model fine-tuned on the EM task does not change if the process starts from a model pre-trained on sentence similarity as the one offered by SBERT [39]. Ditto [24], Supcon [22, 36], i.e., two approaches specialized to address the EM task, BERT and SBERT seem to rely on the same form of knowledge. Only SupCon, trained with a contrastive learning technique, seems to be able to differentiate what it learns from matching and non-matching entity descriptions. (5) Fine-tuning BERT models on the EM task via a binary classifier does not allow the model to completely learn that the attributes of the entity descriptions represent a useful semantic context to support model prediction. Improving fine-tuning to learn these distinctive characteristics of the EM process could improve the effectiveness of the approaches.

The rest of the paper is organized as follows: Section 2 introduces related work; in Sect. 3, we define the boundaries

of our experimental evaluation that results in an analysis of the impact of fine-tuning on the effectiveness (Sect. 4); of what BERT learns from the dataset structure (Sect. 5); the importance of the pair-wise semantic similarity of tokens in the EM process (Sect. 6); the impact of pre-train in the accuracy (Sect. 7) and how the fine-tuning technique can learn the EM task (Sect. 8). In Sect. 9, we point out some lessons learned, and in Sect. 10, we sketch out some conclusions and future work. The source code, and the datasets are available at the project github.¹

2 Related work

2.1 BERT architecture overview

BERT [11] is a large transformer-based architecture whose main component is a self-attention module [48]. It takes as input a sequence of token representations, learns the reciprocal attention that each token of the sequence directs toward 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.

2.2 BERT inspection

Recently, a thriving collection of works, identifiable under the term BERTology [40], has inspected the BERT architecture to assess its ability to learn correct linguistic artifacts. A

¹ <https://github.com/softlab-unimore/bert-attention-for-em>

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) [14, 26, 37, 43, 44]. 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) [18, 25]: 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 [44] or perturbed by the evaluation methodology itself [7, 17, 26, 50]. Parallel to these works, several approaches directly inspected the BERT architecture [9, 19, 23]. 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 [15, 19], to the best of our knowledge, this is the first work about Entity Matching.

2.3 Deep entity matching

Deep Learning (DL) models can effectively address EM. DeepER [12] and DeepMatcher [32] 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 [48] 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 [33]. Some examples of BERT-based EM systems are [5, 24, 35]. In [5], 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. Li et al. [24] 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 [35], 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 [45] and a survey on the adoption of DL architectures in EM is available in [2].

3 The experimental analysis

3.1 Methodology

We adopt an experimental methodology to answer the following five research questions. We think that the analysis 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? (Sect. 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? (Sect. 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? (Sect. 6)
4. Which is the impact of the technique adopted to pre-train the transformer on its ability to address EM tasks? (Sect. 7)
5. Which fine-tuning technique allows the model to better learn that the inputs are pairs of entity descriptions? Is the binary classification task, which is usually adopted for fine-tuning BERT, effective for allowing BERT learning the EM task? (Sect. 8)

3.2 Datasets

We performed the experiments against the datasets provided by the Magellan library² 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 with a match label (fifth column). In the experiments that require to train and evaluate the effectiveness of BERT in performing EM, we used the split into training set/valid set/test set provided by the benchmark. The remaining experiments,

² <https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md>

Table 1 Magellan benchmark

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

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. Although the EM task is often imbalanced (there are more non-matching than matching entities), we perform the evaluation with balanced test sets having the same number of matching and non-matching entity descriptions. This allows us to perform bias-free evaluations that do not depend on the data distribution and that focus solely on the matching logic learned from the model. The experiments were all repeated three times and the average value is reported in the paper.

3.3 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. We performed the experiments with different *granularities for the data representation*. In some tests, we evaluated the attention given to *tokens*. 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 model³. 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 [5, 24]⁴. 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*). Table 2 shows a summary of the acronyms used in the specification of the experimental settings.

3.4 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* [11] is a squared

³ 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.

⁴ We used the standard *BertForSequenceClassification* architecture from the *HuggingFace's Transformers* library, where the classifier built on top of BERT is composed by a dropout and a single linear layer that compresses the 768-dimensional vector directly into two outputs corresponding to match and non-match classes.

Table 2 Summary of the dimensions of analysis

Data encoding	SP—Sentence Pair AP—Attribute Pair
Attribute representation	AV—Average attention MA—Maximum attention
Model application	PT—Pre-trained Model FT—Fine-tuned Model

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 Fig. 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).

3.5 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* [24] and [5]). 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 [23]. In particular, the analysis of the role of attention modules in complex models has recently been discussed. [4, 20, 41] 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 nonlinear transformations. Nevertheless, this is a controversial point since other papers, as [47, 49], demonstrated that low correlations happen only in limited conditions.

4 Impact of the fine-tuning 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 Sect. 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 Sect. 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 Sect. 4.3, we evaluate if the fine-tuning impacts on the embeddings of matching and non-matching word pairs.

4.1 Effectiveness

4.1.1 Implementation

We evaluate BERT's ability to solve EM tasks by adding on top of its modules a binary classifier as described in Sect. 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 3, and compared with *DeepMatcher+* (DM+) [32], a reference DL-based EM approach that does not rely on a transformer architecture, and *Ditto* [24], one of the best BERT-based EM approaches.

4.1.2 Discussion

Even if DM+ obtains good results in most the datasets, the BERT-based models outperform them. *Ditto*, which extends BERT with data augmentation techniques and advanced EM data encoding, further improves the results. The average results computed on the largest datasets (i.e., the ones with more than 10k records) show that fine-tuning improves the effectiveness more in dirty (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) than in structured datasets. With reference to sentence pair encoding, fine-tuning improves the performance by around 7% in large structured datasets, and more than 10% in large dirty datasets. This result is consistent with what was also observed in [5, 25] and would suggest that fine-tuning does not learn structural knowledge from the data sources. Finally, data encoding (attribute or sentence pair) does not have on average a real impact on the results.

⁵ In Table 3, we consider the results as published in [24].

⁶ The scores differ from [24, 34] since they are computed with the experimental settings adopted in this paper.

Table 3 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
<i>S-FZ (Fodors-Zagais)</i>	97.67	97.67	100.00	97.67	100.00	97.78
<i>S-DG (DBLP-GoogleScholar)</i>	92.80	92.40	94.92	94.78	94.70	94.97
<i>S-DA (DBLP-ACM)</i>	97.52	97.41	98.42	98.65	98.45	96.86
<i>S-AG(Amazon-Google)</i>	65.19	63.26	70.21	68.52	70.70	75.31
<i>S-WA (Walmart-Amazon)</i>	54.81	59.89	79.79	78.85	73.60	85.40
<i>S-BR (BeerAdvo-RateBeer)</i>	82.76	82.76	77.78	84.85	78.80	90.32
<i>S-IA (iTunes-Amazon)</i>	86.21	85.19	90.00	93.10	91.20	92.31
<i>T-AB (Abt-Buy)</i>	62.35	59.50	81.42	83.51	62.80	87.04
<i>D-IA (iTunes-Amazon)</i>	70.59	84.21	94.74	94.74	79.40	83.64
<i>D-DA (DBLP-ACM)</i>	96.85	96.10	98.43	98.42	98.10	96.65
<i>D-DG (DBLP-GoogleScholar)</i>	91.63	92.27	95.07	94.77	93.80	94.86
<i>D-WA (Walmart-Amazon)</i>	56.60	50.76	79.59	77.33	53.80	87.05
<i>Large Struct. AVG (STD)</i>	77.58 (20.83)	78.24 (19.40)	85.84 (13.19)	85.20 (14.04)	84.36 (14.23)	88.11 (9.89)
<i>Large Dirty AVG (STD)</i>	81.69 (21.89)	79.71 (25.14)	91.03 (10.05)	90.17 (11.27)	81.90 (24.43)	92.85 (5.10)
<i>Overall AVG (STD)</i>	79.58 (16.65)	80.12 (17.03)	88.36 (10.04)	88.77 (9.90)	82.95 (15.40)	90.18 (6.74)

The average on large datasets refers to the sources with more than 10,000 records

4.2 Attention

4.2.1 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 [23] to perform NLP tasks. The cosine similarity between (flattened versions of) the attention heads associated with 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 Fig. 1 for the *settings* = (*SP*, *–*, *PT/FT*).

4.2.2 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 [15, 16, 23, 40].

4.3 Embeddings

4.3.1 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 belonging to matching/non-matching pairs. 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 Fig. 2.

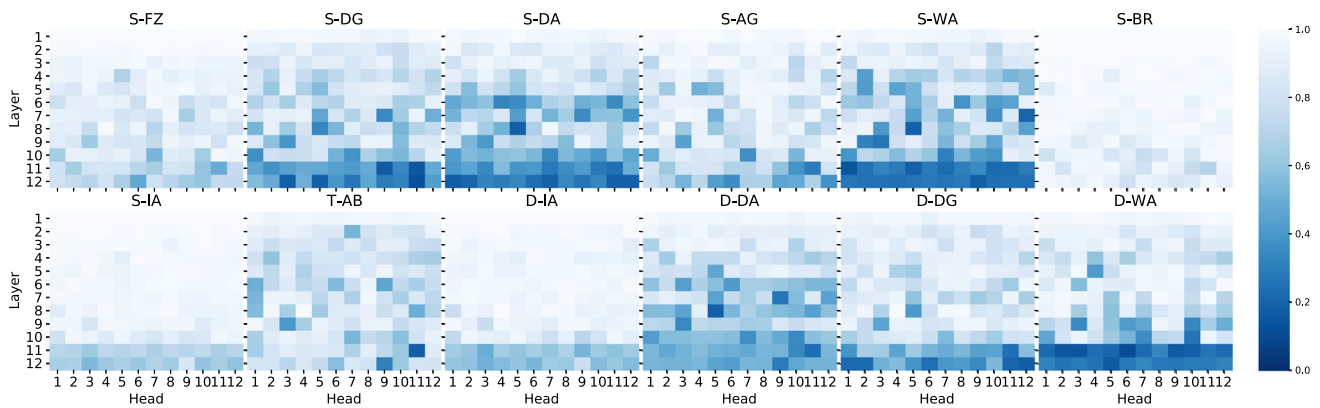


Fig. 1 Similarity between pre-trained and fine-tuned attention scores, $settings = (SP, -, PT/FT)$. The darker the cell, the greater the difference between the attention scores of fine-tuned and pre-trained models

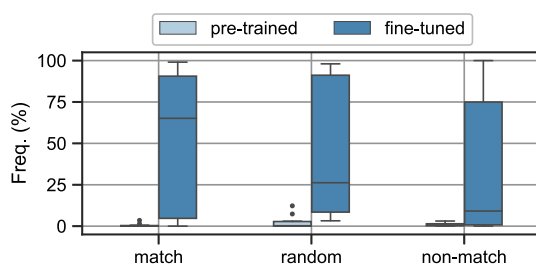


Fig. 2 Impact of the fine-tuning on the similarity of the embeddings: $settings = (SP, -, PT/FT)$. The y-axis shows frequency of the similar embeddings in the entity descriptions

4.3.2 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 Sect. 5.1, we investigate whether and to what extent the fine-tuned

BERT model 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 Sect. 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 Sect. 5.3 evaluate whether 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

5.1.1 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.

5.1.2 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-

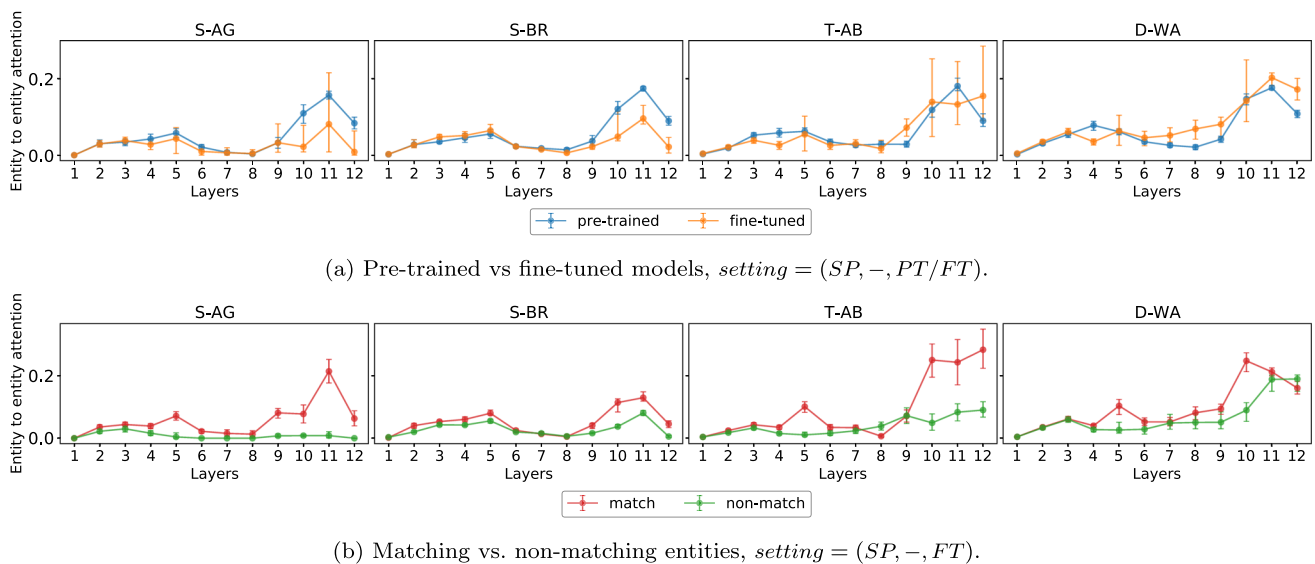


Fig. 3 Entity-to-entity attention. The y-axis reports the normalized number of times where pairs of tokens from descriptions of different entities are assigned an attention score in the last quintile of an average attention head calculated on a certain layer

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 Fig. 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 analyzed.

5.2 The attention patterns involving the dataset attributes

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

5.2.2 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 toward itself and neighboring attributes. (2) *vertical*: vertical lines show that all attributes have high attention toward 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 [23]. The MAA is a new pattern emerging in the EM scenario: an attribute gives high attention toward 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

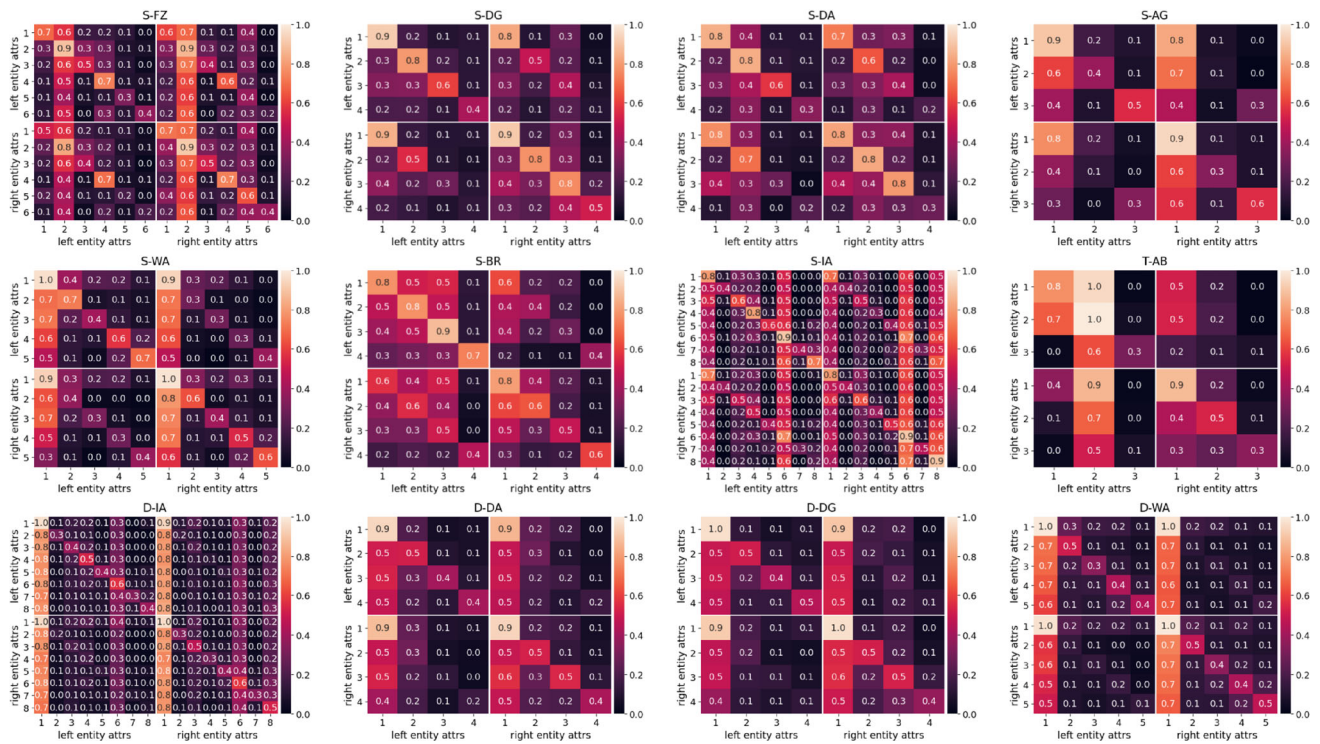


Fig. 4 Attention between attributes: *setting* = (*SP*, *MA*, *PT*). The cells show the attention given by attributes on rows to attributes on columns, divided by left and right entity. The lighter the color, the higher the attention

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.

5.2.3 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 model effectiveness.

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, 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 (impor-

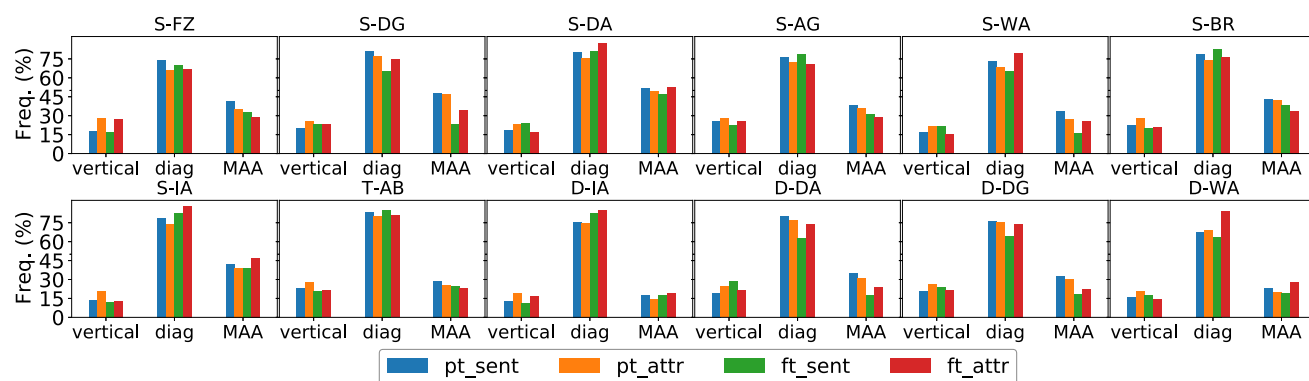


Fig. 5 Comparison of pattern frequency in all the settings (AP/SP , MA , PT/FT). The bars show the percentage of the attribute attention heads where the patterns are found



Fig. 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

tance), we prune the same number of heads, but according to their importance (as defined in [29]). 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 108.5 to 99.6M. The results of this experiment are reported in Fig. 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.

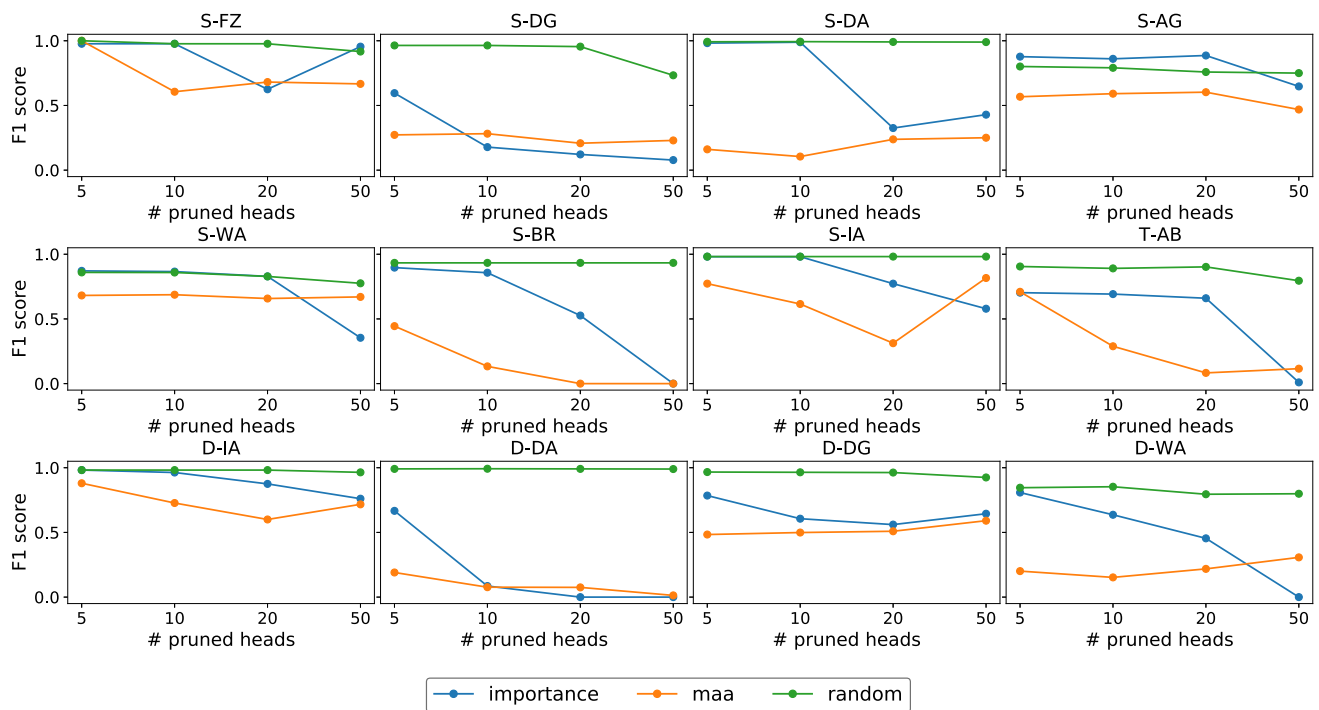


Fig. 7 Impact of the MAA pattern on the effectiveness of the EM task performed with fine-tuned BERT models

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 with 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 toward 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 Fig. 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 with 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 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 toward 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 toward one attribute, while on the structured version to multiple attributes. To elaborate on the analysis, Fig. 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 author 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

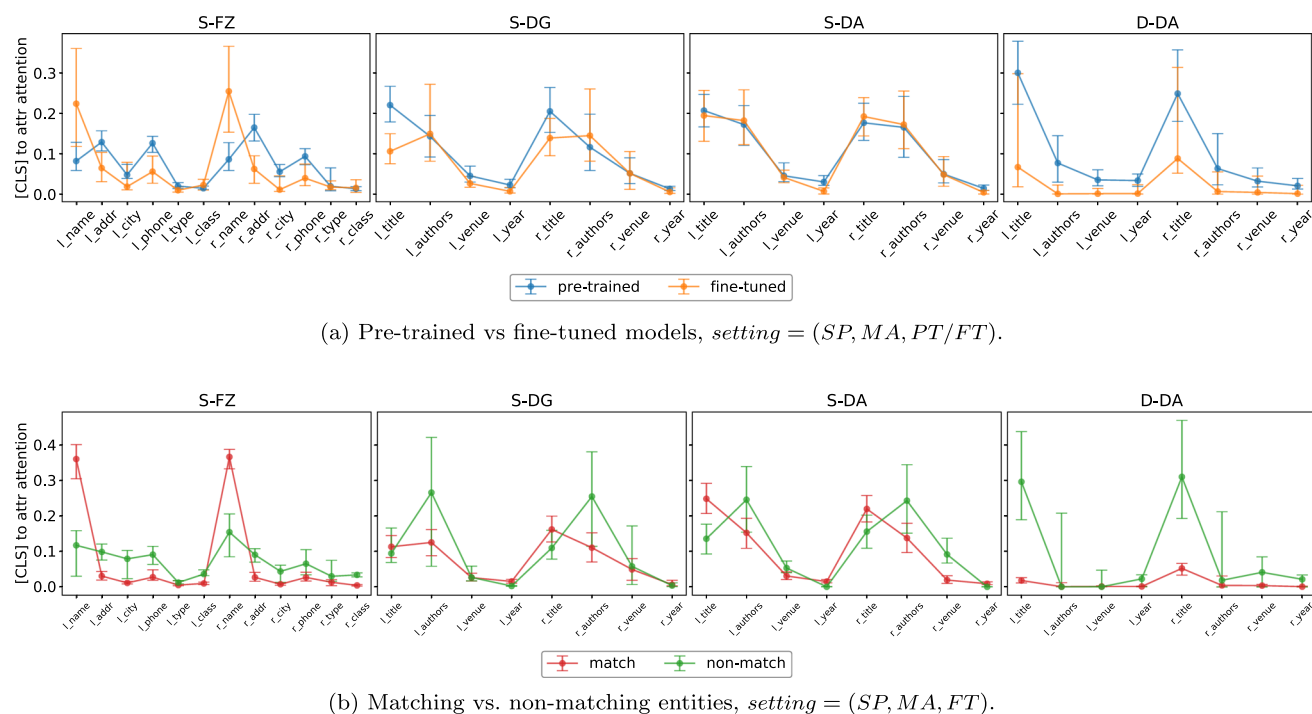


Fig. 8 Attention given by the token [CLS] to the dataset attributes. The diagram shows the average value computed on all dataset entries of the attention registered in the last layer of the attention heads

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 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 [42] 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 Fig. 9.

Discussion We observe that the results are consistent with those obtained in the previous experiment: the first attribute of each dataset contains the most discriminative information for identifying the entities (e.g., the attribute title is the first attribute in dataset S-DG). In the other datasets, attributes either are located in descending order of importance or there is no real difference among their values.

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 [3] and we analyze how BERT treats these inputs. In Sect. 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 Sect. 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 Sect. 6.3, we perform a gradient analysis to evaluate the contribution of the semantic relationships on the inferences.

6.1 Attention and semantic similarity

6.1.1 Implementation

The goal of this experiment is to evaluate the extent of the attention that BERT gives to words with high similar-

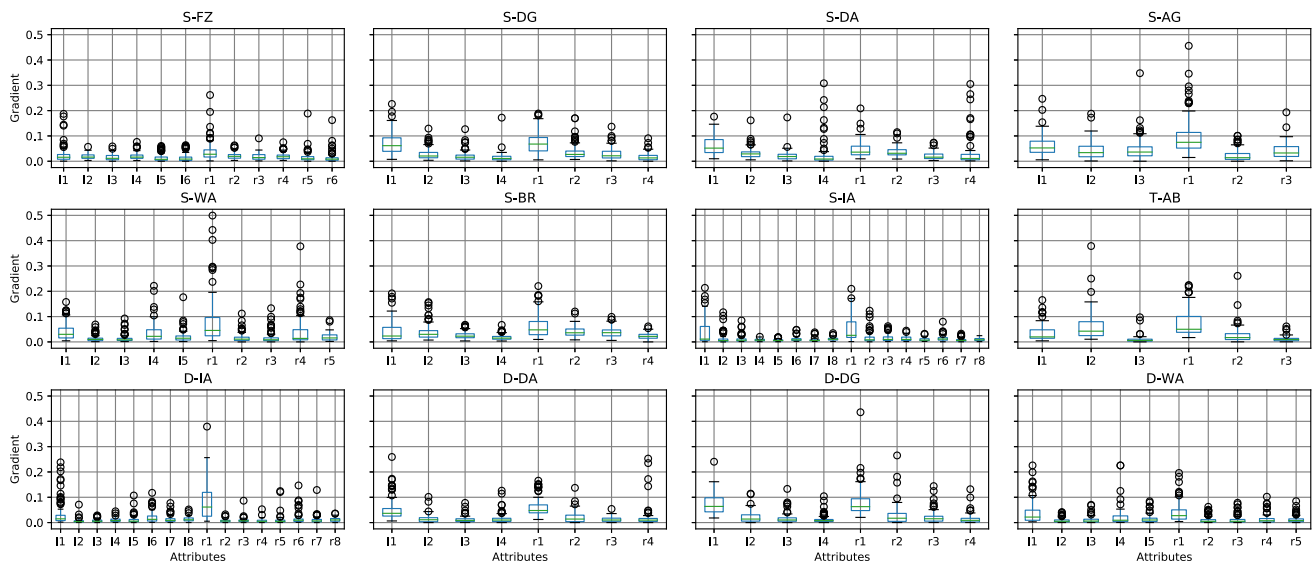
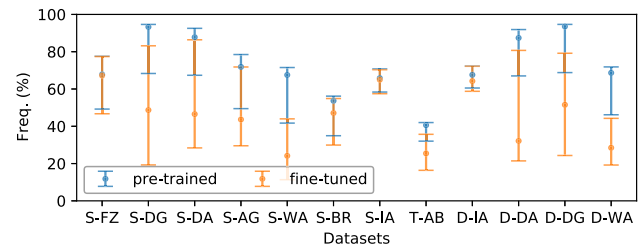


Fig. 9 Importance given to the attribute computed with the gradient analysis. The highest the value, the highest the importance of the attribute for the prediction

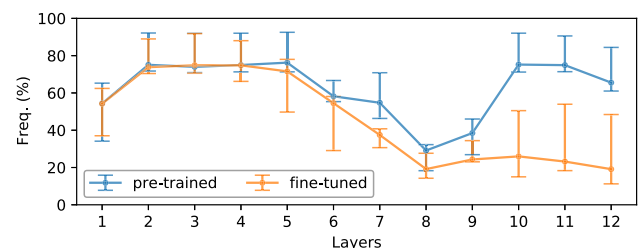
ity in solving the EM task. For each dataset record, we create two sets: one including the pairs of words with the highest attention score, i.e. the ones in the last quartile according to the values computed on an average attention head obtained by averaging all heads referring to the same layer; and 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. Figure 10a and b shows the results of the experiment, aggregated per dataset and per layer, respectively, on the setting = (SP, −, PT/FT).

6.1.2 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 [18, 25] 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 Sect. 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 Sect. 5.2.3).



(a) Frequency across the datasets.



(b) Frequency across the layers.

Fig. 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

6.2 Embeddings and semantic similarity

6.2.1 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 Sect. 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 Fig. 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.

6.2.2 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

6.3.1 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 Sect. 5.3.2, we use the technique described in [42] 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 Fig. 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*).

6.3.2 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 The impact of the pre-training technique

This Section investigates the importance of the technique adopted for pre-training transformer-based models in learning how to solve EM. In the previous Sections, we experimented with the standard BERT model, which is pre-trained to perform two tasks: the prediction of masked words and the prediction of the next sentences. The effectiveness of these techniques has been largely demonstrated in many NLP problems. The Masked Language Model task for example resulted to be extremely competitive in introducing linguistic knowledge into the model [27]. However, we have limited knowledge of whether these pre-training techniques are the most effective to learn EM. We, therefore, wonder if applying a different pre-training technique to the model could improve the accuracy in addressing EM tasks.

We reviewed many transformer-based models proposed in the literature (see [30] for a survey) and decided to experiment with SBERT [39], Ditto, and SupCon [36]. SBERT is a modification of the pre-trained BERT network that uses a Siamese Network to generate semantically meaningful sentence embeddings. Siamese Networks have been adopted for addressing many tasks. Among them, [28] applied (straightforwardly) this architecture to address EM. The resulting model is thus pre-trained on a task close to EM, where the distance between pairs of sentences (and not entity descriptions) is evaluated. Ditto and SupCon are two transformer-based approaches designed, respectively, for entity and product matching. As stated in the related work Section, Ditto is a BERT-based model customized for solving EM by means of the application of domain knowledge and data augmentation to the input data. SupCon [36] is a transformer-based model for product matching that applies a pre-training procedure based on supervised contrastive learning [22]. The idea is to force the model to create embedding representations that are close for descriptions referring to the same real-world entities and are far for different entities.

In this Section, we analyze how fine-tuning to the EM task a different pre-trained transformer-based model, or adopting a BERT-based model customized for the EM: (1) can improve the effectiveness of the models (Sect. 7.1); (2) introduces specific structural knowledge into the models, and, in particular, the notion of matching attributes (Sect. 7.2); (3) introduces syntactic and/or semantic knowledge into the models (Sect. 7.3).

Our goal is to analyze these transformer-based models according to the same perspectives through which the BERT model was analyzed in Sects. 4, 5, and 6. However, reproducing on these models each experiment performed on BERT is not possible. The selected transformer-based models are architecturally different. BERT and Ditto are cross-encoders: the model processes a pair of sentences jointly using a special separator [SEP] to distinguish the sen-

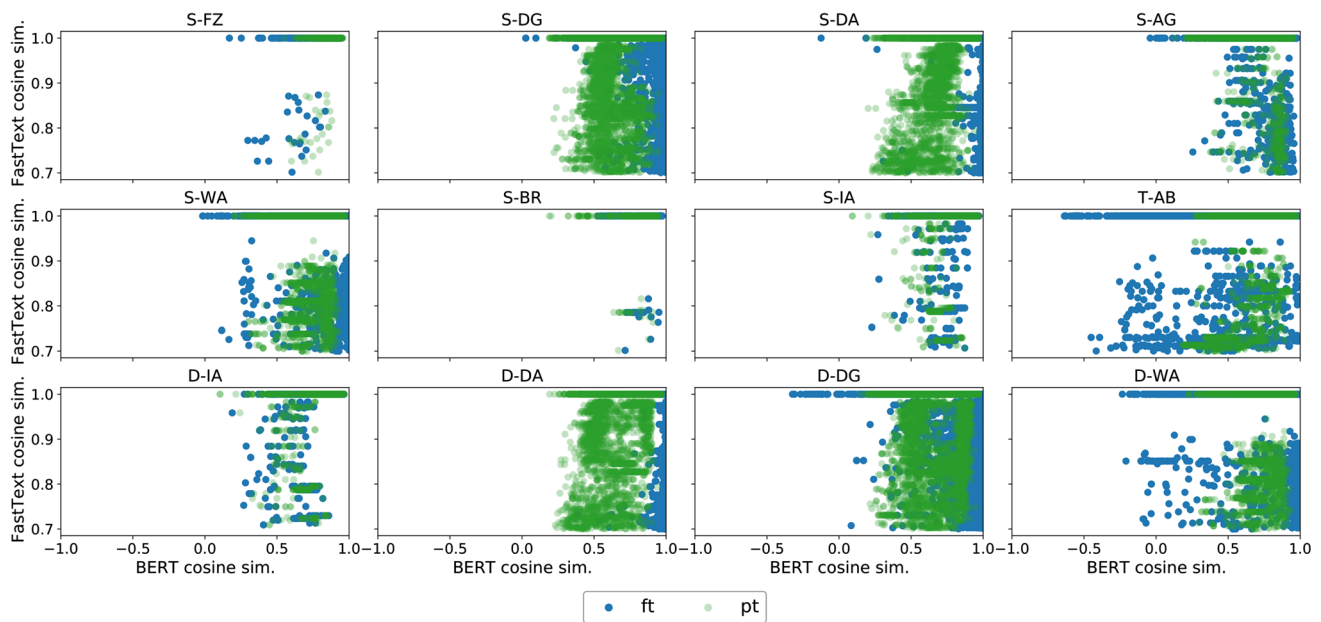


Fig. 11 Comparison between BERT and fastText embeddings. The diagrams show the cosine similarity of the embeddings generated by BERT for word pairs with cosine similarity greater than 0.7 according to the fastText encodings

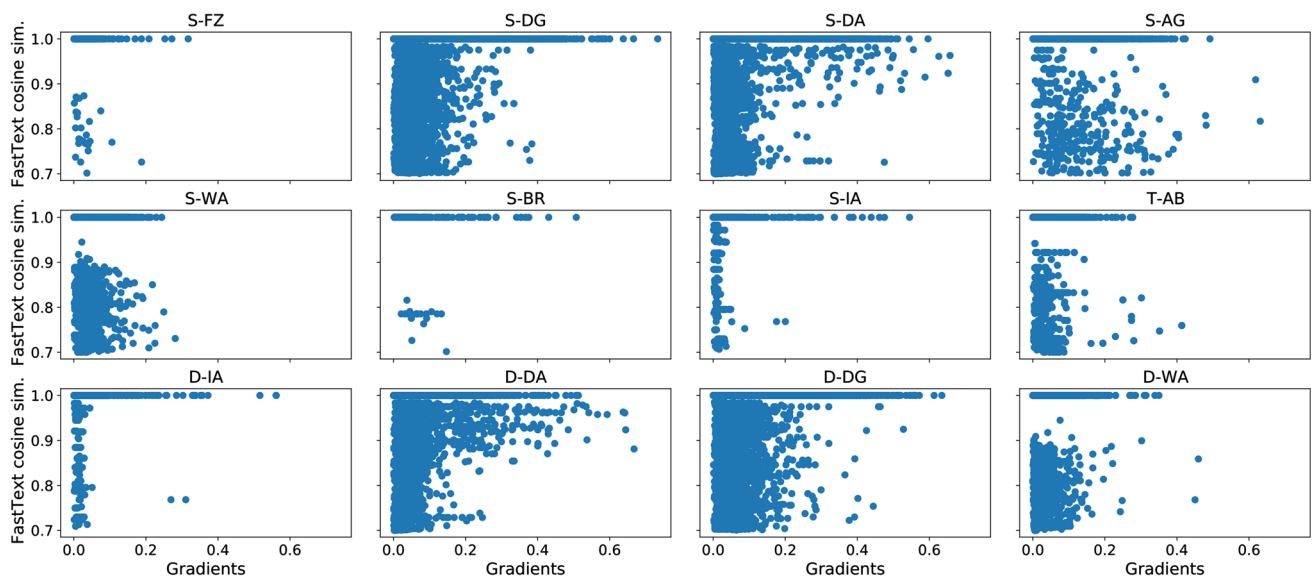


Fig. 12 Comparison between gradient and semantic similarity. The gradients generated by a fine-tuned BERT model are compared with the cosine similarity of the fastText embeddings of pairs of words (only similarities greater than 0.7 are shown)

tences (and attributes in Ditto). SBERT and SupCon are bi-encoders: two encodings, one for each input sentence, are independently generated and used for the prediction. Since bi-encoders process the sentences as a whole and cannot detect interactions between words from different sentences, the comparison is performed through the following subset of experiments that evaluate the models as “black box applications”, i.e., by evaluating the outputs to given inputs only.

7.1 Effectiveness in performing EM

7.1.1 Implementation

The aim of the experiment is to evaluate the effectiveness of differently pre-trained transformer-based models in executing the EM task. As we did in Sect. 4.1 with BERT, we evaluate SBERT with four configurations, by analyzing the impact of the data encoding (attribute and sentence pair) and the fine-tuning, i.e., (SP/AP , $-$, PT/FT). Ditto and Sup-

Table 4 The effectiveness of fine-tuned SBERT, Ditto and SupCon: $settings = (SP/AP, -, PT/FT)$ (F1 score)

	Pre-trained (attr-pair)	Pre-trained (sent-pair)	Fine-tuned (attr-pair)	Fine-tuned (sent-pair)	Ditto	SupCon
<i>S-FZ (Fodors-Zagats)</i>	93.33	97.67	100.00	100.00	97.78	92.68
<i>S-DG (DBLP-GoogleScholar)</i>	92.41	92.47	94.31	94.24	94.97	80.54
<i>S-DA (DBLP-ACM)</i>	96.47	96.84	98.76	98.30	96.86	99.21
<i>S-AG (Amazon-Google)</i>	63.72	64.88	68.84	60.48	75.31	79.23
<i>S-WA (Walmart-Amazon)</i>	59.62	60.23	78.22	78.05	85.40	80.12
<i>S-BR (BeerAdvo-RateBeer)</i>	85.71	82.76	73.68	84.85	90.32	96.55
<i>S-IA (iTunes-Amazon)</i>	75.00	77.19	81.82	93.10	92.31	85.71
<i>T-AB (Abt-Buy)</i>	58.52	57.79	84.26	84.18	87.04	93.43
<i>D-IA (iTunes-Amazon)</i>	60.71	75.00	91.53	93.10	83.64	68.18
<i>D-DA (DBLP-ACM)</i>	96.83	95.98	98.54	98.42	96.65	99.44
<i>D-DG (DBLP-GoogleScholar)</i>	91.11	91.22	94.80	95.05	94.86	80.13
<i>D-WA (Walmart-Amazon)</i>	56.14	55.26	76.32	76.68	87.05	77.06
<i>Large Struct. AVG (STD)</i>	78.06 (19.07)	78.61 (18.72)	85.03 (13.94)	82.77 (17.24)	88.14 (9.91)	84.79 (9.64)
<i>Large Dirty AVG (STD)</i>	81.36 (22.03)	80.82 (22.26)	89.89 (11.90)	90.05 (11.70)	92.85 (5.10)	85.54 (12.13)
<i>Overall AVG (STD)</i>	77.46 (16.71)	78.94 (16.19)	86.76 (10.90)	88.04 (11.71)	90.18 (6.74)	86.02 (10.04)

The average on large datasets refers to the sources with more than 10,000 records

Con are executed with the standard configurations introduced in their presentation papers (see the project github for more details). Table 4 shows the accuracy (in terms of F1 score) achieved by these models. The experiment reproduces with these pre-trained models which in Sect. 4.1 is experimented with the BERT model.

7.1.2 Discussion

The comparison of Tables 3 and 4 (describing BERT) shows that BERT and SBERT achieve similar accuracy levels in almost all datasets. The reason for this could be imputed to the benchmarking datasets analyzed, where it is easy for the models to perform well and then there is not a big diversity of results. The only datasets where we observe a marked difference are S-IA and D-IA where the pre-trained version of SBERT performs better than BERT of about 10%. Nevertheless, after the fine-tuning, the difference between the approaches is really close. Moreover, SBERT shows the same behavior observed in Sect. 4.1 for BERT: fine-tuning improves on average the performance more for dirty than for structured datasets. For example, if we consider attribute-pair encodings, fine-tuning improves the performance by 9.23% in dirty datasets and 4.16% in structured datasets. Ditto achieves the best effectiveness: it obtains an average F1 score of 90.18%, which is higher by 2–4 points than the other tested models. This derives from the injection of domain knowledge and the application of a more advanced technique for encoding attribute values. Finally, we observe that the average performance of SupCon is not good as expected. In some datasets (e.g., T-AB, D-DA, S-DA, S-BR, and S-AG), it outperforms Ditto on average of about 4%. In structured and dirty DBLP-GoogleScholar and iTunes-Amazon, it obtains very poor performance (on average 12.5% lower). One of the reasons is that the approach was executed with the standard hyper-parameters, with no specific fine-tuning for the selected datasets.

7.2 Impact on the structural knowledge

7.2.1 Implementation

The experiment evaluates the impact of the fine-tuning process on pre-trained transformer-based models in learning the existence of matching attributes in the pairs of entity descriptions. This is an important knowledge characterizing the EM task, where the pairs of entity descriptions are typically structured in attributes. Since the attributes divide the entity descriptions in semantic contexts, acquiring this kind of knowledge can improve the model effectiveness and interpretability. In Sect. 5, we observed how this knowledge is detected by the BERT model. The goal of the experiment is to investigate if the BERT, SBERT, Ditto, and SupCon mod-

Table 5 Structural statistics of GMASK groups

	BERT				SBERT			
	Entropy (non-match)	Entropy (match)	Matching ratio (non-match)	Matching ratio (match)	Entropy (non-match)	Entropy (match)	Matching ratio (non-match)	Matching ratio (match)
S-FZ	0.67 ± 0.13	0.72 ± 0.09	0.31 ± 0.12	0.25 ± 0.11	0.65 ± 0.11	0.67 ± 0.12	0.33 ± 0.13	0.30 ± 0.14
S-DG	0.72 ± 0.15	0.63 ± 0.12	0.34 ± 0.15	0.40 ± 0.14	0.72 ± 0.14	0.65 ± 0.14	0.35 ± 0.15	0.41 ± 0.16
S-DA	0.67 ± 0.15	0.63 ± 0.16	0.37 ± 0.16	0.44 ± 0.18	0.68 ± 0.12	0.61 ± 0.16	0.38 ± 0.14	0.43 ± 0.20
S-AG	0.59 ± 0.14	0.53 ± 0.18	0.59 ± 0.19	0.60 ± 0.20	0.57 ± 0.16	0.55 ± 0.15	0.57 ± 0.20	0.56 ± 0.20
S-WA	0.64 ± 0.11	0.54 ± 0.14	0.32 ± 0.12	0.34 ± 0.13	0.55 ± 0.15	0.55 ± 0.14	0.41 ± 0.12	0.34 ± 0.14
S-BR	0.76 ± 0.14	0.72 ± 0.15	0.08 ± 0.07	0.06 ± 0.08	0.78 ± 0.12	0.75 ± 0.13	0.11 ± 0.06	0.04 ± 0.09
S-IA	0.53 ± 0.10	0.51 ± 0.09	0.01 ± 0.05	0.03 ± 0.03	0.55 ± 0.08	0.53 ± 0.09	0.02 ± 0.04	0.03 ± 0.05
T-AB	0.59 ± 0.16	0.59 ± 0.14	0.17 ± 0.20	0.19 ± 0.22	0.65 ± 0.17	0.63 ± 0.15	0.15 ± 0.19	0.20 ± 0.23
D-IA	0.41 ± 0.14	0.44 ± 0.12	0.00 ± 0.01	0.01 ± 0.02	0.40 ± 0.14	0.39 ± 0.11	0.00 ± 0.02	0.01 ± 0.02
D-DA	0.46 ± 0.19	0.48 ± 0.18	0.45 ± 0.24	0.48 ± 0.25	0.46 ± 0.21	0.41 ± 0.20	0.40 ± 0.20	0.49 ± 0.23
D-DG	0.39 ± 0.21	0.36 ± 0.17	0.41 ± 0.18	0.48 ± 0.19	0.50 ± 0.21	0.47 ± 0.18	0.45 ± 0.23	0.46 ± 0.17
D-WA	0.46 ± 0.15	0.38 ± 0.17	0.39 ± 0.15	0.44 ± 0.16	0.40 ± 0.18	0.43 ± 0.17	0.44 ± 0.16	0.38 ± 0.18
AVG	0.57	0.54	0.28	0.31	0.57	0.55	0.30	0.31
STD	0.12	0.11	0.18	0.19	0.12	0.11	0.18	0.18
	Ditto				SupCon			
	Entropy (non-match)	Entropy (match)	Matching ratio (non-match)	Matching ratio (match)	Entropy (non-match)	Entropy (match)	Matching ratio (non-match)	Matching ratio (match)
S-FZ	0.63 ± 0.07	0.62 ± 0.06	0.20 ± 0.08	0.19 ± 0.09	0.77 ± 0.10	0.00 ± 0.00	0.18 ± 0.10	0.00 ± 0.00
S-DG	0.47 ± 0.10	0.48 ± 0.11	0.28 ± 0.11	0.30 ± 0.09	0.63 ± 0.16	0.61 ± 0.13	0.28 ± 0.16	0.24 ± 0.17
S-DA	0.44 ± 0.12	0.44 ± 0.12	0.29 ± 0.13	0.32 ± 0.09	0.62 ± 0.15	0.62 ± 0.15	0.28 ± 0.14	0.32 ± 0.19
S-AG	0.46 ± 0.12	0.51 ± 0.16	0.54 ± 0.19	0.51 ± 0.16	0.48 ± 0.12	0.53 ± 0.14	0.54 ± 0.18	0.52 ± 0.17
S-WA	0.45 ± 0.11	0.46 ± 0.01	0.22 ± 0.13	0.22 ± 0.12	0.60 ± 0.11	0.65 ± 0.14	0.18 ± 0.12	0.17 ± 0.10
S-BR	0.50 ± 0.11	0.48 ± 0.09	0.33 ± 0.08	0.31 ± 0.11	0.68 ± 0.13	0.00 ± 0.00	0.31 ± 0.18	0.00 ± 0.00
S-IA	0.56 ± 0.09	0.55 ± 0.07	0.10 ± 0.07	0.11 ± 0.06	0.64 ± 0.09	0.44 ± 0.11	0.09 ± 0.07	0.09 ± 0.06
T-AB	0.49 ± 0.13	0.44 ± 0.12	0.33 ± 0.28	0.00 ± 0.00	0.53 ± 0.18	0.46 ± 0.16	0.43 ± 0.26	0.20 ± 0.22
D-IA	0.34 ± 0.09	0.38 ± 0.09	0.04 ± 0.05	0.04 ± 0.05	0.36 ± 0.12	0.33 ± 0.13	0.17 ± 0.12	0.18 ± 0.05
D-DA	0.40 ± 0.21	0.43 ± 0.20	0.44 ± 0.19	0.47 ± 0.22	0.37 ± 0.18	0.37 ± 0.22	0.33 ± 0.19	0.26 ± 0.19
D-DG	0.26 ± 0.14	0.23 ± 0.16	0.34 ± 0.14	0.39 ± 0.15	0.35 ± 0.25	0.39 ± 0.19	0.40 ± 0.25	0.35 ± 0.16
D-WA	0.31 ± 0.10	0.30 ± 0.13	0.28 ± 0.11	0.28 ± 0.13	0.38 ± 0.15	0.42 ± 0.17	0.24 ± 0.19	0.18 ± 0.10
AVG	0.44	0.44	0.28	0.28	0.53	0.40	0.29	0.21
STD	0.10	0.10	0.13	0.16	0.14	0.21	0.13	0.15

els use this knowledge in the prediction. Our idea is to rely for this purpose on an *explainer*, i.e., a tool that can measure the impacts of the features in the decision made by the model [31].

We introduce GMASK [8], a tool to implicitly detect word correlations by grouping correlated words from input text pairs together and measuring their contribution to the corresponding NLP tasks as a whole. The emphasis on word-group significance motivated the choice to use this tool. GMASK does not associate a degree of importance at the word-level (as usually explainers do), but groups the words that contribute in a similar way and assigns to the group an importance degree. In particular, GMASK relies on WMASK [6] to

obtain an importance score for the words in the record. The top- k ⁷ important words are selected from both input entity descriptions to serve as input for GMASK's subsequent clustering and weighting process. GMASK distributes words into clusters on the basis of the correlation of the embeddings. The number of clusters is decided via a heuristic rule and the importance of each group is learned through a training process that involves random masking of word groups. The plan is to apply GMASK using BERT, SBERT, Ditto, and SupCon, by generating a number of groups equal to the number of attributes in the dataset. The analysis of the groups, and,

⁷ The top-10 words were selected in [8]. We considered 20 words, since each record contains two descriptions.

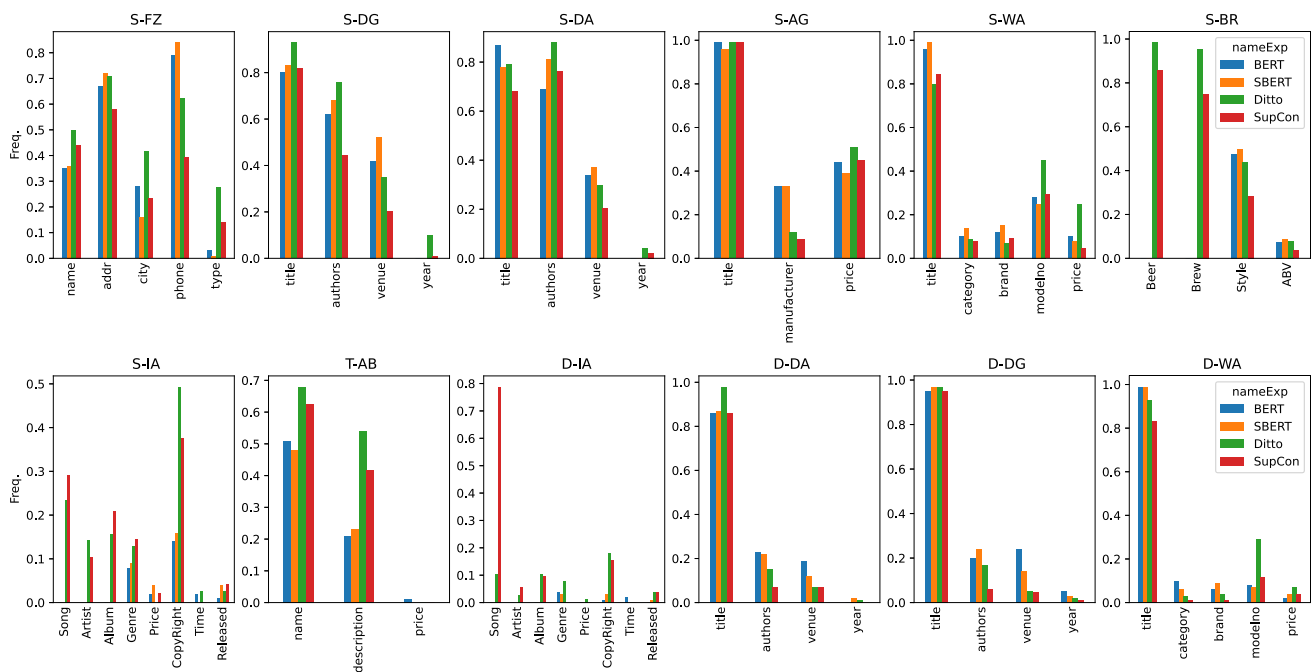


Fig. 13 Provenance of the values in the GMASK groups: the frequency of the values from the same attribute that belong to the same group is shown. Note that to improve the readability, the y-axes have a different graduation

in particular, the measure of their homogeneity will allow us to determine if the models have learned and used the knowledge of the attributes in their predictions. The experiment analyzes a random balanced sample of 100 records and the top 20 words have been selected for grouping. The results are shown in Table 5, where groups generated by GMASK are statistically analyzed in terms of normalized entropy (we calculate how on average the groups are homogeneous in containing values deriving from a few attributes) and matching ratio (the mean of the percentage of values from the same attribute per group). Moreover, Fig. 13 shows the quality of the groups, by reporting the frequency of the values belonging to the same attribute and also to the same GMASK group.

7.2.2 Discussion

Table 5 shows that the application of GMASK generates groups of words with on average similar statistical measures. Moreover, all models do not really make any difference in the number of attributes they base their decision on if they consider matching and non-matching entity descriptions. Only for SupCon, we see a marked difference between the average values of the entropy computed for matching and non-matching descriptions. In particular, the entropy values show that SupCon tends to rely on fewer attributes when dealing with matching than non-matching pairs. Moreover, we observe lower entropy values of dirty than structured datasets: in the presence of noisy information, the models tend to focus more on a smaller number of attributes. The

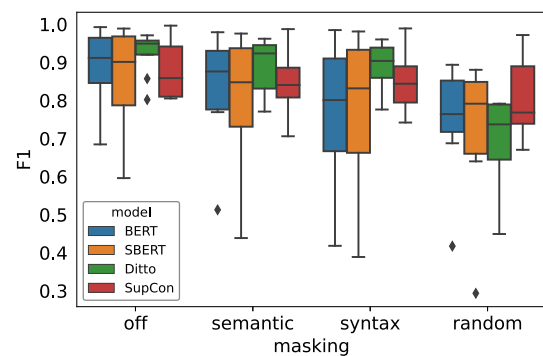


Fig. 14 Accuracy as the masking technique varies (off—no masking, sem—semantic masking, syn—syntactic masking and random masking) with BERT, SBERT, Ditto and SupCon: *settings* = (*AP*, $-$, *PT*) (F1 score)

matching ratio shows that only in a few cases the number of values in a group coming from the same attribute is more than 50% (on average it is around 30%). In addition, Fig. 13 shows that there is no complete correspondence between the groups computed by GMASK on the top words and the attributes in the datasets. This behavior suggests that the importance given by GMASK in explaining the models is partially consistent with the structure of the datasets. However, we observe that in particular for dirty datasets, the first attribute is the one with the largest number of correspondences. This confirms the highest importance given to the first attribute given also by GMASK as emerged for BERT in the experiments of Sect. 5.3.

7.3 Impact on the syntactic/semantic knowledge

This Section introduces two experiments. The first experiment (Sect. 7.3.1) investigates if the fine-tuning allows the SBERT, Ditto and SupCon models to learn syntactic and semantic knowledge about word pairs and how the process is different from BERT as described in Sect. 6. The second experiment (Sect. 7.3.2) compares the embeddings generated by all models for representing the entities.

7.3.1 Learning pair-wise syntactic and semantic knowledge

Implementation In this experiment, the entity descriptions are altered by properly masking some of their words. In particular, we experiment with three techniques: (a) masking of randomly selected words; (b) masking of words with high syntactic similarity (i.e., the ones with edit distance less than 3), and (c) masking of words with high semantic similarity of their corresponding embeddings (i.e., the ones with cosine similarity greater than 0.7). We apply the techniques by masking 3 words from the left and the right descriptions on the 12 datasets used in the experiments. However, in some records, it is not always possible to identify syntactically/semantically close words. For this reason, a sample is extracted for each dataset, containing the entity descriptions that can satisfy all masking techniques. Samples with less than 100 records are removed from the experiments. The resulting masked datasets are as follows: S-DG (3763 records), S-DA (1857), S-AG (1342), S-WA (581), T-AB (1227), D-DA (1534), D-DG (3702) and D-WA (657). Then, we measured the accuracy (F1 score) of the models applied to the masked datasets.

Figure 14 compares the distribution of the F1 score as the masking technique varies (off—without masking, semantic—semantic masking, syntax—syntactic masking and random masking).

Discussion Figure 14 shows that BERT and SBERT reach the same accuracy in all scenarios. We can therefore conclude that the diverse training did not result in the acquisition of different knowledge by the models. Ditto, as expected, shows the best performance (apart from after random masking), and, in particular, the smallest interquartile range, thus demonstrating the low variance of the performance. Moreover, we observe that the average accuracy decreases the most in datasets after the random masking. We could expect less degradation when removal occurs on random words and not on syntactically or semantically related words. Indeed, this is consistent as noted earlier in Sect. 6 with reference to BERT: SBERT, Ditto and SupCon does not rely solely on syntactic/semantic knowledge between single word pairs to effectively perform the EM task. Instead, they seem to rely on a broader and more contextualized knowledge similarly to BERT.

7.3.2 Representing entity descriptions

Implementation We analyze if the transformer-based models learn with the fine-tuning a different way for representing the entity descriptions. For each record, we compute the embeddings of both the entity descriptions⁸ and we calculate the similarity of the pair of embeddings. Table 6 shows the distribution of Jaccard similarities between the entity descriptions divided by matching and non-matching entities. The values in the Table provide a reference for evaluating the similarity of the embeddings. Since the records with pairs of matching entity descriptions have a greater similarity than those with non-matching descriptions, we expect a model that can discriminate between matches and non-matches to encode this “distance” at the level of embeddings. The values of the cosine similarity for the embeddings, grouped by pre-trained and fine-tuned, BERT, SBERT, Ditto and SupCon models are shown in Fig. 15.

Discussion The pre-trained version of the BERT and SBERT shows a compact distribution of the cosine similarity of the embeddings. The fine-tuning operation increases the variability of the cosine similarity values of the embeddings, keeping the median value unchanged. We also observe that BERT and SBERT generate very high cosine similarity values (≥ 0.9) regardless of whether the records refer to matching or non-matching entities. This means that by analyzing the similarity of the embeddings of the descriptions, we cannot discover if the records are describing matching and non-matching entities. This can probably be explained on the basis of the well-known *anisotropy phenomenon*, that makes the token embeddings occupy a narrow cone, resulting in a high similarity between any sentence pair [21]. Ditto shows a similar behavior: the median of the similarity of the descriptions does not significantly change in descriptions referring to matching and non-matching entities. This is expected since Ditto relies on the standard BERT architecture that does not train the model to learn also this kind of knowledge. Conversely, SupCon is the only approach that learns a different behavior for matching and non-matching entity descriptions. The similarity of the generated embeddings is consistent with the Jaccard similarity of the entity descriptions shown in Table 6. This is the result of the contrastive learning technique implemented in SupCon, which requires that descriptions referring to the same entity have closer embeddings than descriptions of different entities.

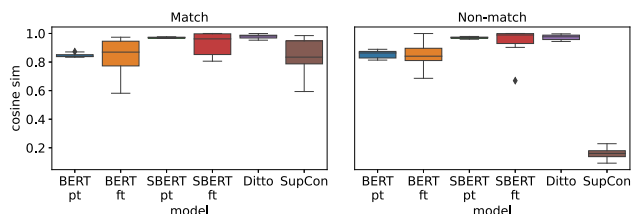
8 Learning the EM tasks

The goal of the experiments in this Section is to understand what BERT has really learned about the EM task. We are

⁸ We average the embeddings of the words in a description.

Table 6 Average Jaccard similarity of the entity descriptions in match and non-match entity descriptions

	Match	Non-match
<i>S-FZ (Fodors-Zagats)</i>	0.569	0.248
<i>S-DG (DBLP-GoogleScholar)</i>	0.538	0.171
<i>S-DA (DBLP-ACM)</i>	0.724	0.149
<i>S-AG (Amazon-Google)</i>	0.423	0.220
<i>S-WA (Walmart-Amazon)</i>	0.407	0.292
<i>S-BR (BeerAdvo-RateBeer)</i>	0.553	0.264
<i>S-IA (iTunes-Amazon)</i>	0.455	0.319
<i>T-AB (Abt-Buy)</i>	0.192	0.133
<i>D-IA (iTunes-Amazon)</i>	0.467	0.333
<i>D-DA (DBLP-ACM)</i>	0.691	0.164
<i>D-DG (DBLP-GoogleScholar)</i>	0.578	0.201
<i>D-WA (Walmart-Amazon)</i>	0.438	0.324
AVG	0.503	0.235
STD	0.141	0.072

**Fig. 15** Sentence similarity in BERT, SBERT, Ditto and SupCon on match and non-match records

interested in finding out if the model learns that records are composed of pairs of descriptions or if it bases its behavior on identifying hidden patterns of co-occurrence between words. The results of the experiments can suggest the need to develop alternative fine-tuning techniques to the usual simple binary classification for increasing the effectiveness of the model.

The experiments in Sect. 8.1 aim to evaluate if and how the dataset structure, composed of attributes, and the domain of the words in the descriptions impact the prediction. Then, the experiment in Sect. 8.2 evaluates the capability of BERT to recognize entities (and not only matching entity descriptions) in the datasets. Finally, two experiments in Sect. 8.3 evaluate the robustness of what the model learned.

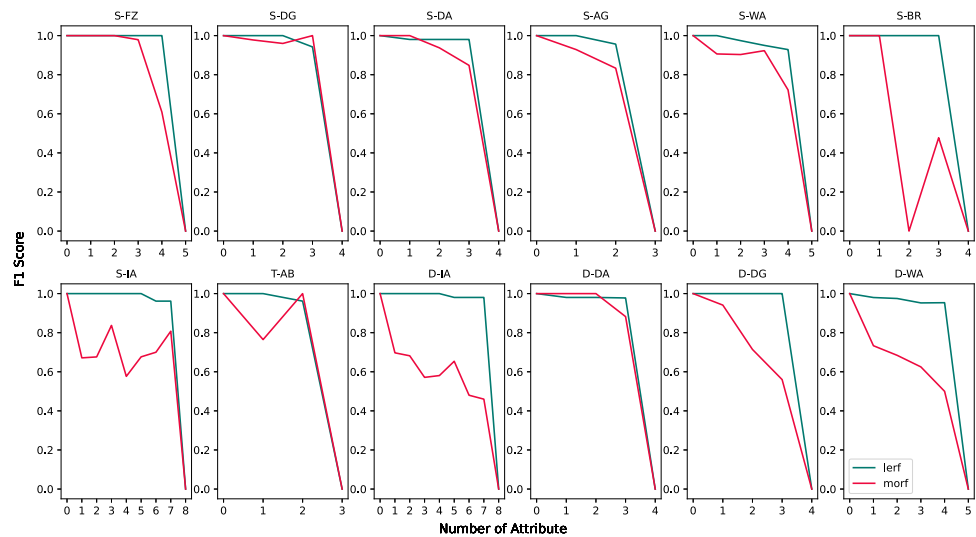
8.1 Exploiting the structure of the entity descriptions and the domains of words in the prediction

The experiments in the Section investigate how much BERT relies on attributes (Sect. 8.1.1) and on word domains (Sect. 8.1.2) to make predictions.

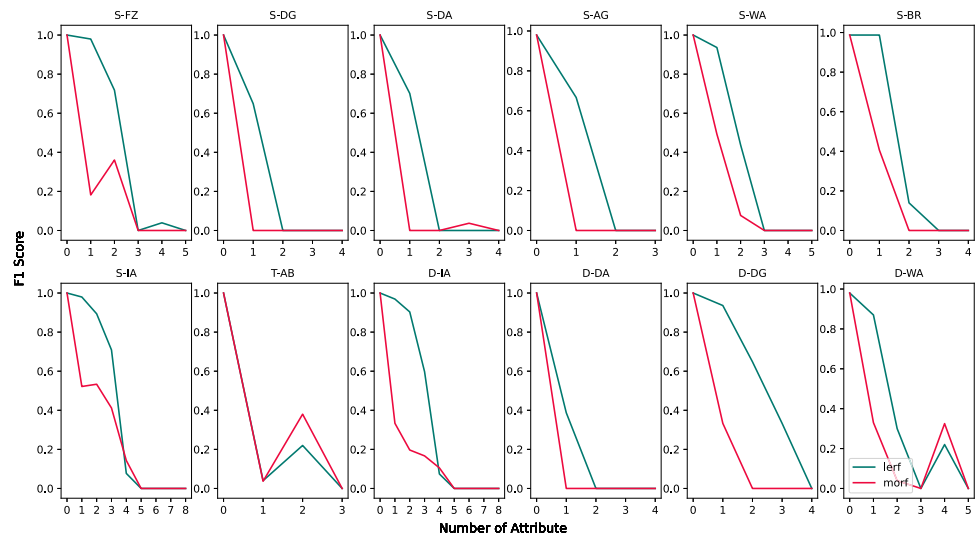
8.1.1 Relying on attributes

Implementation To investigate if BERT relies on the attributes, we perform a degradation test [1] that evaluates the drop in accuracy of the model predictions after the removal of the attributes in the ascending and descending order of their importance. For the sake of simplicity, we consider the attributes as already ordered by importance in the entity descriptions, i.e., the first attribute is more important than the second, and so on. This approximates for all datasets the findings emerged in Sect. 5.3, where we discovered that the first attribute for each description is the most important one (this happens for all datasets) and the other attributes are ordered by descending importance (we find this in the majority of the datasets). Figure 16a shows two curves: one with the most relevant attributes removed first (MoRF) and another one with the least relevant attributes removed first (LeRF). The x-axis shows the number of attributes removed, and the y-axis is the accuracy of the prediction (F1 Score). Figure 16b shows the same experiment but performs a word-level degradation. In this case, the words are first ordered by the importance attributed by WMask and then grouped into a number of groups equal to the number of attributes that composes the entity description. Word groups are considered attributes and removed following the same procedure as in the previous experiment. This represents a baseline where attributes in the descriptions have less importance since the words are partitioned into groups independently of the attributes they belong to. Note that the experiments differ from those in Sect. 5 where attribute importance was assessed as the composition of the impacts of the tokens (e.g., adding up the gradients of the tokens), while here the importance refers to the attribute as a whole.

Discussion Figure 16a shows that the LeRF curve is always above the MoRF and the knowledge provided by the most important attribute is usually enough to guarantee highly accurate predictions. In almost all the datasets, the degradation shown by the MoRF curve in the first steps is not marked. This indicates that even if we remove the most important attribute(s) usually enough knowledge remains to the rest of the entity descriptions for accurately computing the predictions. We hypothesize that BERT does not base the prediction on the knowledge provided by attributes as a whole, but instead bases it on “important words” distributed in different attributes. This hypothesis is confirmed by Fig. 16b which shows the effect of the removal of words for importance regardless of their belonging to particular attributes. The MoRF curves drop very quickly indicating how the removal of the most important words (and not of the attributes) significantly degrades the performance of the model.

Fig. 16 Importance of the attributes on the predictions

(a) Removing the LeRF and MoRF attributes.



(b) Removing the LeRF and MoRF words.

8.1.2 Relying on domain-specific words

Implementation To understand the importance given to the domains of words in the entity descriptions, we perform with BERT explained by GMASK the experiment proposed in [35]. In particular, the words in the entity descriptions are manually classified into 7 classes (i.e., model number, brand name, model name, characteristic attribute, stop word, product type, and other descriptive word) and GMASK is applied to compute the top-20 words along with their importance scores. Figure 17 shows the results of the experiments where the selected words are aggregated per class. The experiment is performed on a sample of the WDC benchmark [38] collecting computer offers to compare the results with the ones obtained analyzing BERT explained by LIME [35].

Discussion The experiment shows a high variance in the importance of the words in the categories. If we consider the median values, model name, brand name, and model number are the top-3 word classes where BERT relies on. The result is consistent with the human perception of the indicators for matching or non-matching product offers. The result is also consistent both in terms of variance and median values with [38], where the same top-3 classes are computed even with more emphasis on the importance of the model number and name than on the brand name.

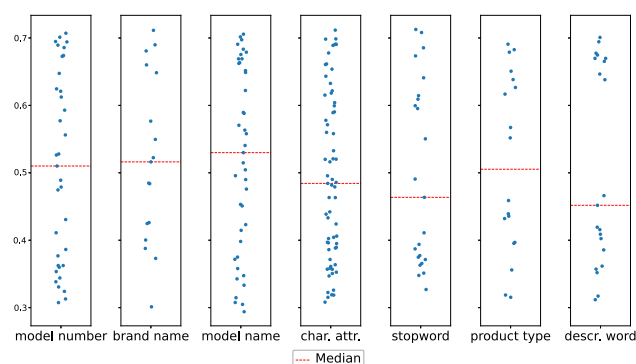


Fig. 17 Importance to word domains attributed by GMask in the prediction

8.2 Recognizing the entities

8.2.1 Implementation

In this experiment, we evaluate the ability of a BERT-based EM model to perform *Entity Resolution* (i.e., to identify groups of descriptions that refer to the same real-world entity). Real-world entities are typically identified by computing the transitive closure process of matching entity descriptions. Only when a clique is formed, an entity representing the matching descriptions is found [13]. The EM task is usually modeled in the literature as a binary classification problem. Therefore, the EM model cannot recognize multiple pairs of entity descriptions referring to the same real entity. Nevertheless, evaluating if the BERT-based EM model can preserve the cliques provides us insights into its understanding of the entity concept.

We examine how many cliques obtained by applying the transitive closure to the match predictions are recognized by the model with respect to the ground truth. Not all the datasets used in the previous experiments can be used for this experiment. Only the S-DG and D-DG datasets generate cliques of size greater than 2. To provide a more challenging dataset, we also perform the experiment against the categories of computers, video cameras, shoes, and watches from the WDC benchmark.

We train an EM model using the training set from the benchmark, applied the model to the validation set, and calculated the cliques comprising descriptions of matching entities. We also execute the experiments with SupCon, since we observed in Sect. 7.3.2 that the model generates discriminative embeddings, i.e. able to encode the similarity of the entity descriptions. Table 7 shows the results of the experiment. The first column reports the number of cliques in the ground truth. The other columns show the percentage of cliques not correctly recognized by the models and the accuracy obtained in terms of F1 score.

8.2.2 Discussion

Table 7 shows that an average of 16% of cliques are not recognized by the BERT model, even if the model reaches a high level of accuracy (more than 92% on average). The results of the experiment align with what we obtained in Sect. 7.3.2: if the model had been able to recognize entities it could have generated distinct embeddings for each entity instead of generating high similarity between each pair of descriptions. A similar result is achieved by SupCon, where the lower level of accuracy impacts the number of cliques found. However, we observe that in datasets where the models have similar effectiveness, SupCon is able to find a larger number of cliques.

8.3 Robustness of learning

The experiments in this Section aim to evaluate the robustness of what the EM models learned by evaluating their predictions to properly modified datasets. In Sect. 8.3.1, we inject repeated words in the entity descriptions to discover if the model is based on co-occurrence patterns. In Sect. 8.3.2, we applied the EM models to datasets different from the ones where they have been trained to evaluate their ability to generalize.

8.3.1 Robustness to spurious co-occurrence patterns

Implementation The experiment investigates whether the matching predictions computed by the BERT-based EM model are the result of poorly interpretable co-occurrence patterns. We alter the test sets of each dataset by selecting only the records referring to non-matching entity descriptions and perturbing them by injecting the same word, extracted randomly from the training set of the same dataset, to the left and right descriptions. We repeat the operation 10 times per record, each time injecting a different randomly selected word. We then apply the EM model (trained on the original training set) to these altered test sets and compute the percentage of matching entities. Our idea is that by injecting random tokens, which are independent of the pairs of descriptions, a model, which does not rely on co-occurrence, should not be able to recognize the descriptions (originally evaluated as non-matching) as matching based on the repetitions of the same token. We prepared 3 perturbed test sets from a dataset, with the random word injected 3, 5, and 10 times, respectively. Table 8 reports the average size of the entity descriptions and the percentage of them which is due to the perturbation. We observe that records are composed of an average of 20 words. By injecting the same word 10 times per entity description, we generate a new dataset where on average half of the words are the results of the perturbation. By injecting 5 words, the perturbed datasets

Table 7 BERT and SupCon accuracy in discovering entities

	# cliques	Uncompleted cliques (%)		F1	
		BERT (%)	SupCon (%)	BERT	SupCon
Computers	83	13.25	10.84	92.82	88.78
Cameras	44	6.82	4.55	90.85	90.25
Shoes	44	20.45	29.55	89.04	74.25
Watches	70	17.14	11.43	94.47	80.95
S-DG (Valid)	50	18.00	34.00	94.78	80.54
D-DG (Valid)	50	24.00	36.00	94.77	80.13
AVG	56.83	16.61	21.06	92.79	82.48

Table 8 Length of the records and size of the perturbations in the three settings of the experiment

Dataset	Avg record length	Perturbation size % (repeat=3)	Perturbation size % (repeat=5)	Perturbation size % (repeat=10)
S-FZ	21.63	13.87	23.12	46.23
S-DG	15.58	19.26	32.09	64.18
S-DA	18.37	16.33	27.22	54.44
S-AG	10.55	28.44	47.39	94.79
S-WA	19.2	15.63	26.04	52.08
S-BR	16.07	18.67	31.11	62.23
S-IA	40.73	7.37	12.28	24.55
T-AB	23.33	12.86	21.43	42.86
D-IA	44.12	6.80	11.33	22.67
D-DA	20.87	14.37	23.96	47.92
D-DG	17.4	17.24	28.74	57.47
D-WA	20.85	14.39	23.98	47.96
AVG	22.39	15.43	25.72	51.45

are composed of about 25% of injected words and by injecting 3 words, the perturbation covers about 15% of the record length. The results of this experiment are reported in Fig. 18, where for each dataset, the distribution of prediction changes is reported.

Discussion Figure 18 shows that for some datasets (in particular, S-BR, S-IA, D-IA—the smallest datasets—and D-WA where the interquartile range spans a large interval) the results are very sensitive to the injected words. This generates blurred results that cannot be interpreted. Excluding these datasets, we observe that on average the percentage of change is less than 25%, and the result is not largely affected by the number of repeated words (only for S-DG the median significantly changes). These results seem therefore to suggest that somehow the models are robust to spurious co-occurrence patterns.

8.3.2 Robustness to out-of-distribution records

Implementation In this experiment, we aim to evaluate the robustness of EM models against out-of-distribution data, i.e., their behavior with data that differs from the training set. The experiment is inspired by [46], which explores domain

adaptation techniques for deep EM models. Following a similar experimental evaluation, we evaluate four EM approaches (BERT, SBERT, Ditto, and SupCon) against two scenarios. In the first scenario, we experiment with the models in test sets from the same domain as the training data. For instance, we train the EM models with S-WA and we evaluate them against T-AB, since both datasets describe products. The second scenario, on the other hand, evaluates the performance of models where the training sets and the test sets are from different domains. For example, S-IA collects songs and D-DA publications. Table 9 shows the results of the experiments.

Discussion In the first scenario, where we consider datasets for training and testing belonging to the same scenario, we observe that the EM models exhibit high performance reaching an average F1 score in the range of 0.75–0.80. For the datasets S-DA and D-DA, the scores are really close to the ones achieved with the training and testing set from the same dataset (see Table 3). The poorest results concern the experiments involving T-AB. This dataset is structurally different from S-WA even if it belongs to the same domains (it includes large textual attributes). In the second scenario where training and test datasets are from

Fig. 18 Prediction changes in non-matching records by injecting 3, 5 and 10 times the random word

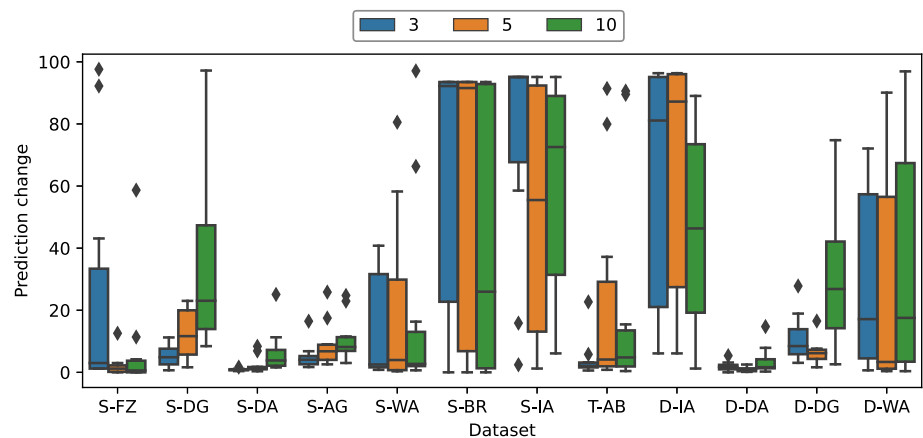


Table 9 Robustness to out-of-distribution records *settings* = (*SP*, $-$, *FT*)

Domain	Source	Target	BERT	SBERT	Ditto	SupCon
Same	S-WA	T-AB	0.50	0.48	0.53	0.33
	T-AB	S-WA	0.46	0.51	0.56	0.39
	S-DG	S-DA	0.95	0.96	0.92	0.96
	S-DA	S-DG	0.75	0.70	0.87	0.92
	D-DG	S-DA	0.96	0.95	0.94	0.96
	D-DG	D-DA	0.96	0.95	0.95	0.96
		AVG	0.76	0.76	0.80	0.75
Different	S-IA	S-DA	0.39	0.53	0.32	0.91
	S-IA	S-DG	0.37	0.46	0.31	0.78
	S-DA	S-IA	0.60	0.64	0.84	0.71
	S-DG	S-IA	0.83	0.79	0.45	0.75
	D-IA	D-DA	0.48	0.64	0.16	0.81
	D-DA	D-IA	0.55	0.67	0.72	0.61
		AVG	0.54	0.62	0.47	0.76
		AVG	0.65	0.69	0.63	0.76

different domains, the performance decreases for all models apart from SupCon. This could be the result of the contrastive learning technique implemented in the model which makes the approach able to better generalize than the learning techniques.

9 Lessons learned

Summarizing the results obtained from the experiments, we observe that even if BERT-based architectures represent a breakthrough in performing EM (Sect. 4.1), the reasons why they largely support the process can be only partially explained. Answering the five questions introduced in Sect. 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 Sect. 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 [15, 16, 23, 40]) 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 (Sect. 4.3).

(2) *The specific structure of the EM datasets as composed of descriptions of pairs of matching / non-matching entities is recognized 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 (Sect. 5.1) but also that matching attributes are recognized (Sect. 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 (Sect. 5.2.3). 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 (Sect. 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 (Sect. 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 (Sect. 6.3). The model seems to focus on a more contextualized kind of knowledge where pragmatic knowledge (discovered by BERT) complements the semantic knowledge.

(4) *Different forms of pre-training allow the model to achieve a different performance in the effectiveness, that is not generally motivated by a different learning of structural, syntactic, and semantic knowledge.* The evaluation of four

BERT models pre-trained with different techniques (i.e., the usual masked-language technique, its improvement for the EM proposed by Ditto with the injection of domain knowledge and structural knowledge, the sentence similarity based offered by SBERT and SupCon based on contrastive learning to deal with product matching) allows us to discover that (a) the effectiveness of BERT and SBERT does not vary significantly and is lower than the one achieved by the specialized approaches Ditto and SupCon (Sect. 7.1); (b) fine-tuning does not lead BERT and SBERT to learning structural knowledge from the entity descriptions (Sect. 7.2); (c) all approaches do not seem to rely for their decisions on semantic and syntactic knowledge only (Sect. 7.3); (d) SupCon is the only approach that can differentiate the knowledge encoded in the embeddings between matching and non-matching entity descriptions (Sect. 7.3.2).

(5) *The fine-tuning process based on binary classification allows models to be effective but does not allow them to learn and exploit key knowledge for the EM process.* Fine-tuning BERT-based models induces robustness to spurious co-occurrence patterns in the matching process (Sect. 8.3.1), but it does not allow the models to recognize cliques of entity descriptions (Sect. 8.2), to exploit the knowledge derived from the attributes (Sect. 8.1), and limits the generalization capacity of models to out-of-distribution data (Sect. 8.3.2).

10 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 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. Finally, we observed that the accuracy achieved fine-tuning the model on the EM task did not change if we start from SBERT, a transformer pre-trained on sentence similarity, i.e., a task close to EM. The Ditto and SupCon approaches, customized for the EM task, are more effective but confirm the emerged findings from the other transformer-based approaches. This led us to conclude that the development and testing of specialized fine-tuning techniques on the EM task will allow transformers to effectively use the structural knowledge of the dataset records. This knowledge is at the moment certainly learned from the model but is not being adequately used.

Future work will be devoted to further clarifying the reasons that make this architecture so performing on the EM task by identifying the components that contribute most to the matching predictions and experimenting with different fine-

tuning approaches. In particular, we think that pre-trained models on sentence-embedding similarity could be evaluated for this purpose. Our idea is to develop and experiment with “entity-aware” fine-tuning techniques, where the goal is the generation of similar embeddings for all entity descriptions involved in the clique.

References

1. Ancona, M., Ceolini, E., Öztireli, A.C., Gross, M.H.: A unified view of gradient-based attribution methods for deep neural networks. CoRR abs/1711.06104 (2017)
2. Barlaug, N., Gulla, J.A.: Neural networks for entity matching: a survey. ACM Trans. Knowl. Discov. Data **15**(3), 52:1–52:37 (2021)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
4. Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M., Wattenhofer, R.: On identifiability in transformers. In: ICLR, Open-Review.net (2020)
5. Brunner, U., Stockinger, K.: Entity matching with transformer architectures - a step forward in data integration. In: EDBT, Open-Proceedings.org, pp. 463–473 (2020)
6. Cao, N.D., Schlichtkrull, M.S., Aziz, W., Titov, I.: How do decisions emerge across layers in neural models? interpretation with differentiable masking. In: EMNLP (1), Association for Computational Linguistics, pp. 3243–3255 (2020)
7. Cao, S., Sanh, V., Rush, A.M.: Low-complexity probing via finding subnetworks. CoRR abs/2104.03514 (2021)
8. Chen, H., Feng, S., Ganhotra, J., Wan, H., Gunasekara, R.C., Joshi, S., Ji, Y.: Explaining neural network predictions on sentence pairs via learning word-group masks. In: NAACL-HLT, Association for Computational Linguistics, pp. 3917–3930 (2021)
9. Clark, K., Khandelwal, U., Levy, O., Manning, C.D.: What does BERT look at? an analysis of bert's attention. In: Black-boxNLP@ACL, Association for Computational Linguistics, pp. 276–286 (2019)
10. Dai, Y., de Kamps, M., Sharoff, S.: BERTology for machine translation: what BERT knows about linguistic difficulties for translation. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, pp. 6674–6690, <https://aclanthology.org/2022.lrec-1.719> (2022)
11. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1), Association for Computational Linguistics, pp. 4171–4186 (2019)
12. Ebraheem, M., Thirumuruganathan, S., Joty, S.R., Ouzzani, M., Tang, N.: Distributed representations of tuples for entity resolution. Proc. VLDB Endow. **11**(11), 1454–1467 (2018)
13. Firmani, D., Saha, B., Srivastava, D.: Online entity resolution using an oracle. Proc. VLDB Endow. **9**(5), 384–395 (2016)
14. Goldberg, Y.: Assessing bert's syntactic abilities. CoRR abs/1901.05287 (2019)
15. Hao, Y., Dong, L., Wei, F., Xu, K.: Visualizing and understanding the effectiveness of BERT. In: EMNLP/IJCNLP (1), Association for Computational Linguistics, pp. 4141–4150 (2019)
16. Hao, Y., Dong, L., Wei, F., Xu, K.: Investigating learning dynamics of BERT fine-tuning. In: AACL/IJCNLP, Association for Computational Linguistics, pp. 87–92 (2020)

17. Hewitt, J., Liang, P.: Designing and interpreting probes with control tasks. In: EMNLP/ IJCNLP (1), Association for Computational Linguistics, pp. 2733–2743 (2019)
18. Hewitt, J., Manning, C.D.: A structural probe for finding syntax in word representations. In: NAACL-HLT (1), Association for Computational Linguistics, pp. 4129–4138 (2019)
19. Htut, P.M., Phang, J., Bordia, S., Bowman, S.R.: Do attention heads in BERT track syntactic dependencies?. CoRR abs/1911.12246 (2019)
20. Jain, S., Wallace, B.C.: Attention is not explanation. CoRR abs/1902.10186 (2019)
21. Jiang, T., Huang, S., Zhang, Z., Wang, D., Zhuang, F., Wei, F., Huang, H., Zhang, L., Zhang, Q.: Promptbert: improving BERT sentence embeddings with prompts. CoRR abs/2201.04337 (2022)
22. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: NeurIPS (2020)
23. Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A.: Revealing the dark secrets of BERT. In: EMNLP/IJCNLP (1), Association for Computational Linguistics, pp. 4364–4373 (2019)
24. Li, Y., Li, J., Suhara, Y., Doan, A., Tan, W.: Deep entity matching with pre-trained language models. Proc. VLDB Endow. **14**(1), 50–60 (2020)
25. Lin, Y., Tan, Y.C., Frank, R.: Open sesame: getting inside bert's linguistic knowledge. CoRR abs/1906.01698 (2019)
26. Liu, N.F., Gardner, M., Belinkov, Y., Peters, M.E., Smith, N.A.: Linguistic knowledge and transferability of contextual representations. In: NAACL-HLT (1), Association for Computational Linguistics, pp. 1073–1094 (2019)
27. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: a robustly optimized BERT pretraining approach. CoRR abs/1907.11692 (2019)
28. Loster, M., Koumarelas, I.K., Naumann, F.: Knowledge transfer for entity resolution via large pre-trained language models: a survey. ACM J. Data Inf. Qual. **13**(1), 2:1–2:25 (2021)
29. Michel, P., Levy, O., Neubig, G.: Are sixteen heads really better than one?. In: NeurIPS, pp. 14014–14024 (2019)
30. Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I., Roth, D.: Recent advances in natural language processing via large pre-trained language models: a survey. CoRR abs/2111.01243 (2021)
31. Molnar, C.: Interpretable machine learning (ebook). <https://christophm.github.io/interpretable-ml-book> (2018)
32. Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., Raghavendra, V.: Deep learning for entity matching: A design space exploration. In: SIGMOD Conference. ACM, pp. 19–34 (2018)
33. Paganelli, M., Buono, F.D., Pevarello, M., Guerra, F., Vincini, M.: Automated machine learning for entity matching tasks. In: EDBT, OpenProceedings.org, pp. 325–330 (2021)
34. Paganelli, M., Buono, F.D., Baraldi, A., Guerra, F.: Analyzing how BERT performs entity matching. Proc. VLDB Endow. **15**(8), 1726–1738 (2022)
35. Peeters, R., Bizer, C.: Dual-objective fine-tuning of BERT for entity matching. Proc. VLDB Endow. **14**(10), 1913–1921 (2021)
36. Peeters, R., Bizer, C.: Supervised contrastive learning for product matching. In: WWW (Companion Volume). ACM, pp. 248–251 (2022)
37. Peters, M.E., Neumann, M., Zettlemoyer, L., Yih, W.: Dissecting contextual word embeddings: architecture and representation. In: EMNLP, Association for Computational Linguistics, pp. 1499–1509 (2018)
38. Primpeli, A., Peeters, R., Bizer, C.: The WDC training dataset and gold standard for large-scale product matching. In: WWW (Companion Volume). ACM, pp. 381–386 (2019)
39. Reimers, N., Gurevych, I.: Sentence-bert: sentence embeddings using siamese bert-networks. In: EMNLP/IJCNLP (1), Association for Computational Linguistics, pp. 3980–3990 (2019)
40. Rogers, A., Kovaleva, O., Rumshisky, A.: A primer in bertology: what we know about how BERT works. Trans. Assoc. Comput. Linguist. **8**, 842–866 (2020)
41. Serrano, S., Smith, N.A.: Is attention interpretable?. In: ACL (1), Association for Computational Linguistics, pp. 2931–2951 (2019)
42. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. ICML PMLR Proc. Mach. Learn. Res. **70**, 3319–3328 (2017)
43. Tenney, I., Das, D., Pavlick, E.: BERT rediscovers the classical NLP pipeline. In: ACL (1), Association for Computational Linguistics, pp. 4593–4601 (2019)
44. Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R.T., Kim, N., Durme, B.V., Bowman, S.R., Das, D., Pavlick, E.: What do you learn from context? probing for sentence structure in contextualized word representations. In: ICLR (Poster), OpenReview.net (2019)
45. Thirumuruganathan, S., Li, H., Tang, N., Ouzzani, M., Govind, Y., Paulsen, D., Fung, G.M., Doan, A.: Deep learning for blocking in entity matching: a design space exploration. Proc. VLDB Endow. **14**(11), 2459–2472 (2021)
46. Tu, J., Fan, J., Tang, N., Wang, P., Chai, C., Li, G., Fan, R., Du, X.: Domain adaptation for deep entity resolution. In: SIGMOD Conference. ACM, pp. 443–457 (2022)
47. Vashishth, S., Upadhyay, S., Tomar, G.S., Faruqui, M.: Attention interpretability across NLP tasks. CoRR abs/1909.11218 (2019)
48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS, pp. 5998–6008 (2017)
49. Wiegrefe, S., Pinter, Y.: Attention is not not explanation. In: EMNLP/IJCNLP (1), Association for Computational Linguistics, pp. 11–20 (2019)
50. Wu, Z., Chen, Y., Kao, B., Liu, Q.: Perturbed masking: parameter-free probing for analyzing and interpreting BERT. In: ACL, Association for Computational Linguistics, pp. 4166–4176 (2020)
51. Zeakis, A., Papadakis, G., Skoutas, D., Koubarakis, M.: Pre-trained embeddings for entity resolution: an experimental analysis. Proc. VLDB Endow. **16**(9), 2225–2238 (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.