Do Transformer Networks Improve the Discovery of Rules from Text?

Mahdi Rahimi, Mihai Surdeanu
Department of Computer Science
University of Arizona, Tucson, Arizona, USA
{marahimi, msurdeanu}@email.arizona.edu

Abstract

With their Discovery of Inference Rules from Text (DIRT) algorithm, Lin and Pantel (2001) made a seminal contribution to the field of rule acquisition from text, by adapting the distributional hypothesis of Harris (1954) to patterns that model binary relations such as *X treat Y*, where patterns are implemented as syntactic dependency paths. DIRT's relevance is renewed in today's neural era given the recent focus on interpretability in the field of natural language processing. We propose a novel take on the DIRT algorithm, where we implement the distributional hypothesis using the contextualized embeddings provided by BERT, a transformer-network-based language model (Vaswani et al., 2017; Devlin et al., 2018). In particular, we change the similarity measure between pairs of slots (i.e., the set of words matched by a pattern) from the original formula that relies on lexical items to a formula computed using contextualized embeddings. We empirically demonstrate that this new similarity method yields a better implementation of the distributional hypothesis, and this, in turn, yields patterns that outperform the original algorithm in the question answering-based evaluation proposed by Lin and Pantel (2001).

Keywords: Rule Acquisition, Distributional Hypothesis, DIRT

1. Introduction

Lin and Pantel (2001) proposed a method for the acquisition of rules from text through an extension of the distributional hypothesis (Harris, 1954) to rules.[1] That is, "if two paths tend to link the same set of words, ... their meanings are similar." The resulting algorithm, called Discovery of Inference Rules from Text (DIRT), has had a wide impact in natural language processing (NLP). For example, it has been shown that such rules acquisition improves question answering (Ravichandran and Hovy, 2002), information extraction (Shinyama and Sekine, 2006), and textual entailment (Melamud et al., 2013b). For example, the pattern *X treat Y* may be used to extract the answer to the question *Which drugs relieve stomach ache?* (Melamud et al., 2013b). The algorithm itself has been improved and adapted to various scenarios (see the Related Work section for a larger discussion).

Unfortunately, the idea of using such rule acquisition and the DIRT algorithm have largely been forgotten once the "deep learning tsunami" hit NLP around 2015 (Manning, 2015). While neural network have brought unquestionable performance improvements to most NLP tasks (as one example among many, the leader board for TACRED,[2] a popular relation extraction task, is completely dominated by neural methods), some advantages of rule-based approaches have been lost. Most importantly, rule-based models are interpretable. That is, every model decision can be assigned to one or a small number of rules, which, generally, are understandable by humans. Moreover, unlike *post-hoc* explainability methods (Ribeiro et al., 2016; Ribeiro et al., 2018) the interpretability provided by rules is *actionable*, i.e., a human expert can correct a rule that does not perform as intended (Valenzuela-Escárcega et al., 2016). This actionable interpretability mitigates the technical debt of NLP systems (Sculley et al., 2015), which is one of the reasons why they were popular in industry (Chiticariu et al., 2013).

Motivated by these observations, in this paper we aim to combine the advantages of modern neural directions with the benefits provided by rule-based methods. More concretely, we propose a new take on the DIRT algorithm, where we implement the distributional hypothesis using the contextualized embeddings provided by BERT, a transformer-network-based language model (Vaswani et al., 2017; Devlin et al., 2018). In particular, we change the similarity measure between pairs of slots (i.e., the set of words matched by a pattern) from the original formula that relies on lexical items to a formula computed in embedding space. We empirically demonstrate that this new similarity method yields a better implementation of the distributional hypothesis, i.e., we have a better understanding if "two paths tend to link the same set of words."

The key contributions of our work are:

- To our knowledge, this is the first work that combines contextualized embeddings with the discovery of rules. The resulting algorithm, called **B**ERT-**I**nformed **R**ule **D**iscovery (BIRD), aims to marry the advantages of both approaches. That is, we use the capacity of transformer networks to semantically model text, but output interpretable rules, similar to the original DIRT algorithm.

- We reproduce the rule learning evaluation from original DIRT paper (Lin and Pantel, 2001), and

[1] Alternatively called patterns or paths.

[2] https://paperswithcode.com/sota/relation-extraction-on-tacred

showed that BIRD performs better than the original DIRT algorithm in most scenarios. We perform a qualitative analysis of the outputs, which indicates that operating in embedding space generalizes better.

2. Related Work

Rule learning approaches for natural language processing tasks achieved peak popularity in the late 1990s – early 2000s (Yarowsky, 1995; Riloff, 1996; Collins and Singer, 1999; Riloff et al., 1999; Yangarber, 2003; McIntosh and Curran, 2008, inter alia). As mentioned, these directions have the advantage of interpretability that is actionable. However, most of these early approaches were iterative semi-supervised algorithms, i.e., they alternate between learning new rules, and acquiring new training examples for their respective tasks (e.g., pairs of entities for binary relation extraction) from matches of the current set of rules. These iterative strategies have been shown to be prone to semantic drift where "ambiguous or erroneous" information is "introduced in the iterative process" (McIntosh, 2010). In contrast, DIRT implements a single pass algorithm that directly implements the distributional hypothesis for rules (Lin and Pantel, 2001). Despite its simplicity, Lin and Pantel (2001) have shown that DIRT performs well for a complex question answering task.

Following the original algorithm, several extensions to DIRT have been introduced. Dinu and Wang (2009) use a hand-crafted lexical resource to increase the original inference rule collection as well as ruling out some of the incorrect rules. Melamud et al. (2013b) also use lexical expansion; they improve the learning of inference rules between rare patterns by lexically expanding the collection of slot-filler words of paths with semantically similar words. Bhagat et al. (2007) determine the directionality of an inference rule by an algorithm that uses the distributional hypothesis and selectional preferences. Szpektor and Dagan (2008) adapt DIRT for unary patterns, in order to learn unary entailment rules. Ibrahim et al. (2003) propose an approach that applies a modified version of DIRT to the same monolingual parallel corpus used by Barzilay and McKeown (2001). Chklovski and Pantel (2004) extend DIRT to consider also antonyms (i.e., opposite meanings). Sun and Grishman (2010) extend the ideas in DIRT to relation extraction by a clustering approach that uses pattern clusters to guide relation extraction methods. Several previous works have been focused on providing context-sensitive extensions to DIRT, addressing the issue of multiple senses per pattern. Examples include learning selectional preferences (Pantel et al., 2007; Roberto et al., 2007; Szpektor et al., 2008), integrating word-level and topic-level representations (Melamud et al., 2013a), modeling senses as latent variables (Dinu and Lapata, 2010), and using a Deep Belief Network based model as a topic model (Guo et al., 2019). However, to the best of our knowledge, we are the first to adapt DIRT to work with contextualized embeddings.

In our approach, we rely on transformer networks to generate contextualized embeddings (Vaswani et al., 2017). In particular, we use BERT, which is a Transformer encoder stack that is pre-trained on a very large corpus using masked language model and next sentence prediction tasks (Devlin et al., 2018).

Our method is close in spirit to Soares et al. (2019). Similar to our direction, they use transformer networks to propose a novel take on the distributional hypothesis for binary relation extraction. However, there are two key differences between the approach of Soares et al. (2019) and ours. First, they train binary classifiers on top of a transformer encoder. In contrast, we produce a collection of rules that extract and explain the relations of interest. We argue that our method is preferable in scenarios where interpretability is crucial, e.g., legal or medical. Second, Soares et al. (2019) rely on distant supervision to generate training data, i.e., they automatically align a database of entity pairs that match the relation of interest with sentences that contain them. In contrast, our approach relies on a bootstrapping approach, which is more applicable in real-world scenarios where such databases are not available.

Several extensions to BERT have been proposed recently to learn relation classifiers from few training examples (few-shot learning), which is similar to our training process (Gao et al., 2019; Sabo et al., 2021). However, these directions are radically different from this work: they generate transformer variants, which have the same interpretability issues as the original BERT, whereas we produce rules.

3. Approach

3.1. Review of the Original DIRT Algorithm

Since our work is a neural interpretation of the DIRT algorithm, for completeness we begin with an overview of the original algorithm. Informally, DIRT learns inference rules from text such as *"X is the author of Y* $\approx$ *X writes Y"*. Some of these inferences are not exact paraphrases (but are still relevant and potentially useful!) such as *"X is the author of Y* $\approx$ *X is known for Y"*. More formally, DIRT is initialized with the left-hand side of the above inference rule, which is implemented as a syntactic pattern (or path) connecting two concepts (see next subsection for details), and infers one or more possible matches for the right-hand side, where each match is represented as a similar syntactic path. DIRT bootstraps such syntactic paths using an adaptation of Harris' Distributional Hypothesis principle (Harris, 1954) to rules. That is, Lin and Pantel (2001) built upon the original distributional hypothesis, which states that words that occur in the same contexts tend to have similar meanings, by rephrasing it for rules: "if two paths tend to link the same sets of words, we hypothesize that their meanings are similar." For example, the three example paths above are likely

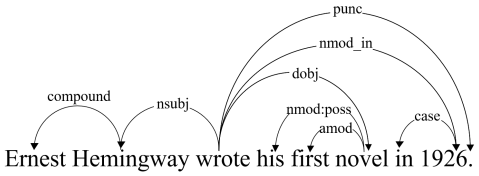

Figure 1: Example dependency tree, which matches the syntactic path nsubj←write→dobj.

to extract similar pairs of concepts such as <"F. Scott Fitzgerald," "The Great Gatsby">.

3.1.1. Extraction of paths:

DIRT extracts syntactic paths from dependency trees. Figure 1 shows an example dependency tree generated by the Stanford CoreNLP dependency parser (Manning et al., 2014). DIRT defines a syntactic path as a concatenation of connected dependency relations and words, but the words at the two ends of the path are excluded. For example, for the sentence in Figure 1, the path between "Hemingway" and "novel" is nsubj←write→dobj. The variables at the two ends of a path are called *slots*: we use SlotX to indicate the slot on the left-hand side of the path, and SlotY the slot on the right-hand side. Slots can be single- or multi-word noun phrases.

3.1.2. Similarity between two paths:

As we are extracting paths, we record and update the frequency of occurrences of each noun phrase as a slot-filler of a path. For each instance of a path *p* with w_1 as the SlotX filler and w_2 as the SlotY filler, we increase the frequency count of two triples *(p, SlotX, w_1)* and *(p, SlotY, w_2)*. In this context, *(SlotX, w_1)* and *(SlotY, w_2)* are called features of the path *p*.

The similarity between two paths is computed based on the intuition that the more slot fillers two paths share, the more similar they are. However, not all words are equally significant. DIRT considers this fact by computing the mutual information between a slot w and a path p:

$$mi(p, slot, w) = log(\frac{|p, slot, w| \times |*, slot, *|}{|p, slot, *| \times |*, slot, w|}) \quad (1)$$

where $|p, slot, w|$ denotes the frequency count of the triple *(p,slot,w)*, $|p, slot, *|$ denotes $\sum_w |p, slot, w|$, and $|*, *, *|$ denotes $\sum_{p,slot,w} |p, slot, w|$.

The similarity between a pair of slots is defined as:

$$sim(slot_1, slot_2) = \frac{\sum_{w \in T(p_1,s) \cap T(p_2,s)} mi(p_1,s,w)+mi(p_2,s,w)}{\sum_{w \in T(p_1,s)} mi(p_1,s,w)+\sum_{w \in T(p_2,s)} mi(p_2,s,w)} \quad (2)$$

In the above equation, $slot_1$ and $slot_2$ are the same type of slots (SlotX or SlotY) of two different paths, $p1$ and $p2$, and $T(p_i, s)$ is the set of the slot-fillers for the s slot of path p_i.

Finally, the similarity between two paths p_1 and p_2 is defined as the geometric mean of the similarities of their left and right slots:

$$S(p_1, p_2) = \sqrt{sim(SlotX_1, SlotX_2) \times sim(SlotY_1, SlotY_2)} \quad (3)$$

3.1.3. Searching for the most similar paths:

Given a path, the goal of the DIRT algorithm is discovering the most similar paths to it according to the (revised) distributional hypothesis. However, computing the similarity of the input path with *all* of the extracted paths in a corpus is infeasible. Therefore, the algorithm makes the search space smaller, by first selecting a set of candidate paths with an inexpensive heuristic. A candidate path is defined as a path that has at least one common slot filler with the input path. Next, for each candidate path, the number of shared slot fillers with the input path is counted and the paths with the number of the shared fillers less than a fixed percentage of the total number of unique slot fillers for the input and candidate paths are filtered out. DIRT used 1% for this threshold.

The key building block in DIRT is Equation 2, which measures the similarity of two slots in different paths. The numerator of this equation requires that the two paths share slot fillers that are *lexically identical*. This is an important limitation: two paths that populate slots with fillers that are semantically similar but lexically distinct, e.g., one path extracts "F. Scott Fitzgerald" while another extracts "Francis Scott Key Fitzgerald," will be considered to have a similarity of 0. Our approach mitigates this problem by computing the similarity of slots in the semantic space produced by transformer networks' contextualized embeddings.

3.2. DIRT with Contextualized Embeddings

In this subsection, we detail our method called BIRD (**B**ERT-**I**nformed **R**ule **D**iscovery). In particular, we extend DIRT by introducing two path similarity measures which are computed using the contextualized embeddings provided by BERT. Similar to DIRT, we start the rule discovery with the phase in which we extract paths. This phase is identical to DIRT. However, we compute the similarity between a pair of paths differently. We introduce two algorithms for computing the similarity between a pair of paths: Unweighted BIRD and Weighted BIRD.

3.2.1. Unweighted BIRD:

We first compute an embedding vector for each of the slots of a given path *p*. In order to achieve this goal, we feed each sentence in which the path *p* was observed in the corpus during the path extraction phase into a BERT model as shown in Figure 2. The BERT model does not have a head (e.g., a fully connected layer) on top, which means it outputs the final hidden states for each input token. We use the hidden states corresponding to the SlotX and SlotY filler tokens to generate slot

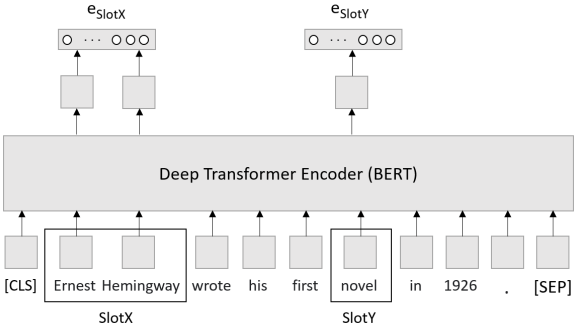

Figure 2: Each sentence of a path is fed to a BERT model in order to obtain contextual embeddings of slot-fillers (e_{SlotX} and e_{SlotY}). Slot-fillers can be single-word or multi-word noun phrases. In the example sentence, SlotX contains two words (`Ernest Hemingway`) and SlotY contains a single word (`novel`).

embeddings. If the filler of a slot is a multi-token expression rather than a single token, we compute and keep the average of the embeddings of the multi-token phrase. Once we have computed all of the individual slot-filler embeddings, we compute the embedding of a slot of a path as the average of all individual slot-filler embeddings:

$$E_s = \frac{1}{n}\sum_i e_{s_i} \quad (4)$$

where e_{s_i} denotes the individual embedding for the filler of slot s in the ith sentence of a certain path, n is the number of sentences where the path matches, and E_s denotes the embedding of the slot s of the path.
Then, the similarity between a pair of slots is defined as:

$$sim(slot_1, slot_2) = \textit{cosine similarity}(E_{s_1}, E_{s_2}) \quad (5)$$

where $slot_1$ and $slot_2$ refer to the same slot type (SlotX or SlotY) of p_1 and p_2 paths. Further, E_{s_1} and E_{s_2} refer to the embeddings of this slot for p_1 and p_2.
Finally, the similarity between a pair of paths p_1 and p_2 is defined as the arithmetic mean[3] of the similarities of their SlotX and SlotY slots:

$$sim(p_1, p_2) = \frac{sim(SlotX_1, SlotX_2) + sim(SlotY_1, SlotY_2)}{2} \quad (6)$$

3.2.2. Weighted BIRD:

Weighted BIRD differs from Unweighted BIRD in how the embedding of a slot of a path is computed. While having more common slot-fillers generally implies more similarities between a pair of paths, the original DIRT algorithm hypothesized that not all common slot-fillers are equally important and influential. For instance, let us assume that we are computing the similarity between the path *"X writes Y"* and another path. Further, let us assume that "novel" and "it" are two common slot-filler words for the SlotY.[4] Because the word "it" is much more frequent than the word "novel", its informativeness is likely to be reduced. Following this observation, we employ the mutual information introduced in Equation 1 as the weights [5] for the individual embeddings of the slot-fillers of a path when we compute the embedding of a slot of a path. Hence, we revisit Equation 4 and redefine a slot embedding as the weighted average of the individual embeddings where the weights are the mutual information between the path slot and its slot-fillers:

$$E_s = \sum_i mi(p, s, w_i) \times e_{s_i} \quad (7)$$

4. Experimental Results

4.1. Experimental Settings

For reproducibility purposes, we reimplemented the original DIRT evaluation as closely as possible. Lin and Pantel (2001) built their evaluation using the first six questions from the TREC-8 Question Answering Track for evaluation. We list these questions in Table 1. TREC (Text REtrieval Conference) is an ongoing annual workshops with the purpose of supporting and promoting research within the information retrieval community. Its Question Answering Track aimed at answering natural language questions, such as those in Table 1 (Voorhees and others, 1999).
The goal of this evaluation is to learn syntactic paths that may answer each of these questions. To this end, each question is transformed into a DIRT/BIRD seed path manually. These paths are shown in column four in the table; the third column lists their English representations. We had to slightly modify Q2 and Q5 in order to extract useful paths from them as it is not possible to express them using a single, contiguous path in the dependency tree. The original TREC Q2 was "What was the monetary value of the Nobel Peace Prize in 1989?". The pattern "X is monetary value of Y" cannot be represented via a contiguous syntactic path due to the modifier "monetary," which is not part of the path that connects X and Y. To address this, we replaced "monetary value" with "price" in Q2, so it can be represented with the path "X is price of Y". Similarly, we modified Q5 by replacing "managing director" with "director."
For this evaluation, we manually verified the top 40 most similar paths for each of the paths in the fourth column of Table 1, via DIRT, Unweighted BIRD, and

[3]We chose the arithmetic mean here because the cosine similarity values may be negative, which would break the geometric mean used in the original DIRT algorithm.

[4]For example, the following two sentences would yield these values for SlotY: "Hemingway wrote his first novel in 1926," and "Hemingway wrote it in 1926".

[5]Mutual information scores can be negative, but this is fine as we can do algebraic operations on embeddings.

Q#	Question	English Representation of Path	Path
Q1	Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?	X is author of Y	nsubj←author→nmod_of
Q2	What was the price of the Nobel Peace Prize in 1989?	X is price of Y	nsubj←price→nmod_of
Q3	What does the Peugeot company manufacture?	X manufactures Y	nsubj←manufacture→dobj
Q4	How much did Mercury spend on advertising in 1993?	X spends Y	nsubj←spend→dobj
		spends X on Y	dobj←spend→nmod_on
Q4e	How much did Mercury spend on advertising in 1993? How much time did the average person spend on social media in 2018? How much energy did the company spend on the project?	spends X on Y	dobj←spend→nmod_on
Q5	What is the name of the director of Apricot Computer?	X is director of Y	nsubj←director→nmod_of
Q6	Why did David Koresh ask the FBI for a word processor?	X asks Y	nsubj←ask→dobj
		asks X for Y	dobj←ask→nmod_for
		X asks for Y	nsubj←ask→nmod_for

Table 1: The first six TREC-8 questions used for evaluation. Each question is accompanied by its corresponding syntactic path that becomes the seed path for that question.

Weighted BIRD algorithms. Lin and Pantel (2001) accept a found path as "correct" if it is possible to create a sentence with the path to answer the question which is being evaluated. For example, let us assume that one of the found paths for Q1 is "X writes Y". If a sentence containing this path could be used to answer Q1 then we will accept the path. In this case, it is possible to create such a sentence (e.g., "Hugo Young wrote the book") and therefore we judge the path as correct.

Not all found paths deemed as correct are strict paraphrases of the queried path. For example, "X is known for Y" can be used to answer Q1 (e.g., "Hugo Young is known for the book "The Iron Lady: A Biography of Margaret Thatcher"), but the path is not a strict paraphrase of "X is author of Y". Lin and Pantel (2001) show some leniency when evaluating the found paths. For example, they judge paths such as "X edits Y" or "X translates Y" as correct for Q1. For these reasons, we also add a new extra criterion for evaluation of paths by manually judging whether the found paths are strict paraphrases of the queried path. Obviously, the set of strict paraphrases is a subset of the set of correct paths.

All these annotations were performed by two annotators (the authors). The Kappa inter-annotator agreement was 56%, which is considered moderate (Landis and Koch, 1977). This is encouraging considering the complexity of the task, and the fact that rules were evaluated out of context, i.e., without access to sentences where they match.

We performed our evaluation with a corpus of 100,000 randomly chosen English Wikipedia articles. We created our corpus from Tensorflow Wikipedia Dataset (TFDS Team, 2021) which contains all of the Wikipedia articles.

For the implementation of BIRD, we used the BERT model of Hugging Face Transformers library (Wolf et al., 2020). We used the cased $\text{BERT}_{\text{BASE}}$ model as it matched both our cased data as well as the compute resources available to us. The Hugging Face BERT model accepts sentences with a maximum length of 512 tokens. For this reason, as well as for reducing computational costs, we discarded sentences with more than 512 tokens in the corpus. We also observed that very long sentences tend to be noisy, so discarding those sentences can potentially improve the results as well. We discarded these sentences for both of BIRD and DIRT implementations.

4.2. Results

The results of the evaluation are presented in Table 2. Each row in the table corresponds to a seed path for one of the TREC questions; the Path column matches the Path column in Table 1. Note that a couple of questions have more than one seed path and, thus, are listed multiple times in the table (once per seed path). For example, we used three seed paths for Q6, similar to Lin and Pantel (2001). Also, because the verb "spend" in Q4 has multiple senses, e.g., "spending time" is different from "spending money," we evaluated Q4.2 twice: the first time keeping the original word sense for the verb (spending money), and the second time (Q4.2e) allowing any sense of the verb as correct.

For each path, we found the top 40 most similar paths using the three approaches discussed: Unweighted BIRD, Weighted BIRD, and our implementation of DIRT. For each approach, we used two criteria for evaluation: we counted the number of "correct" found paths as well as the number of "strict paraphrases"

Q#	Path	Unweighted BIRD (out of 40)		Weighted BIRD (out of 40)		DIRT (out of 40)	
		Correct	Strict Paraphrase	Correct	Strict Paraphrase	Correct	Strict Paraphrase
Q1	X is author of Y	23	10	**24**	**11**	20	5
Q2	X is price of Y	18	9	**21**	**10**	9	3
Q3	X manufactures Y	32	10	**33**	**12**	30	8
Q4.1	X spends Y	7	0	**9**	0	**9**	**2**
Q4.2	spends X on Y	13	6	12	6	**19**	**8**
Q4.2e	spends X on Y	**25**	**14**	24	11	**25**	11
Q5	X is director of Y	**17**	11	**17**	**12**	16	11
Q6.1	X asks Y	15	**2**	**16**	**2**	8	1
Q6.2	asks X for Y	**14**	**7**	13	6	4	2
Q6.3	X asks for Y	21	2	**25**	3	15	**5**

Table 2: Evaluation results for the two BIRD variants compared against the original DIRT algorithm (our implementation).

Q#	Learned Path	English Representation	Example Sentence of the Learned Path from Corpus
Q1	nsubj←write→dobj	X writes Y	Johnston wrote a diary chronicling his activities in the war.
	nsubj←writer→nmod_of	X is writer of Y	Thomas Adolphus Trollope was an English writer of over 60 books.
	nmod:poss←book←nsubj	Y is X's book	Pignat's latest book is a picture book of acrostic poetry about trees.
	nsubj←author→dobj	X authors Y	In later years, Almond authored eight novels in the Alford Saga.
Q2	nmod_at←sell→dobj	sells Y at X	Molycorp, Inc. sold 28,125,000 shares at $14 in its IPO.
	acl→pay→nmod_for	X paid for Y	It was the highest price paid for a motorcycle at auction at that time.
	nsubjpass←give→nmod_for	X is given for Y	A budget is given for the total cost of the solution.
	compound←value→nmod_of	Y value of X	In around 1820 the part of the Marshal-land had a rent value of £58.

Table 3: Example paths learned by BIRD for the first two TREC-8 questions.

found.

4.3. Discussion

Table 2 shows that, overall, BIRD outperforms the original DIRT. Table 3 provides examples of the paths learned by BIRD. In several cases, when BIRD performs better than DIRT, it performs nearly twice better (Q6.3), twice better (Q2 and Q6.1), or more than three times better (Q6.2). In general, BIRD outperforms DIRT in all of the questions except in Q4, which we discuss later in this subsection. Further, Weighted BIRD performs slightly better than Unweighted BIRD in most of the questions. We believe this is due to the fact that the mutual information provides additional insight into the importance of the contextualized embeddings associated with the corresponding slot fillers. We also observed that Unweighted and Weighted BIRD have a considerable overlap between their found paths as depicted in Table 4. However, both BIRD variants have a low overlap with DIRT. This implies that one could further improve results with an ensemble of BIRD and DIRT methods.

We believe BIRD performs better than DIRT for three reasons. First, because it operates in a semantic space that does not require exact lexical match between slot fillers, BIRD suffers less from lexical sparsity than the original DIRT algorithm. This allows it to learn relevant patterns and examples that are more different lexically. Second, BIRD is powered by a transformer network architecture, which takes advantage of the attention mechanism. The attention mechanism looks at the entire input sequence, and creates an embedding representation for the slot-filler tokens holistically which captures the meanings of the slot fillers more precisely by looking at their context. On the other hand, DIRT pays attention solely to the slot-filler tokens, missing important contextual information. Third, BERT's powerful pretraining allows it to provide high-quality and accurate embeddings for slot-fillers, which further contributes to BIRD's superior performance.

On the other hand, DIRT outperformed BIRD in Q4. Q4 is focused on the verb "spend" which has three main meanings: spending time, spending money, and spending energy/effort. However, the TREC question asks

Q#	U. BIRD ∩ W. BIRD		U. BIRD ∩ DIRT		W. BIRD ∩ DIRT	
	Correct	Strict Paraphrase	Correct	Strict Paraphrase	Correct	Strict Paraphrase
Q1	21	10	8	4	8	4
Q2	14	7	1	1	1	1
Q3	27	9	10	1	10	2
Q4.1	6	0	3	0	4	0
Q4.2	10	6	5	3	4	3
Q4.2e	19	11	6	5	5	4
Q5	15	11	3	3	4	3
Q6.1	14	2	3	1	3	1
Q6.2	12	6	2	0	1	0
Q6.3	19	2	8	2	9	3

Table 4: The pairwise intersections of the paths found by Unweighted BIRD, Weighted BIRD, and DIRT in the evaluation.

about spending money on advertising and the evaluation measure accepts a found path as correct only if it is possible to use the path to answer this particular TREC question. This effectively rejects most of the learned paths that are about "spending time". We inspected all of the instances of the path *"X spends Y"* in our corpus. We noticed that 88% of the instances have the meaning of spending time and only 10% have the meaning of spending money. However, it is not always possible to detect the meaning of the verb by looking only at the slot-fillers. For example, in the sentence *he spent the majority of his career in England*, the SlotY filler is "majority" or in the sentence *He spent a great deal of time in Hong Kong*, the SlotY filler is "deal". This class of words (majority, deal, rest, portion, bulk, etc.) look neutral to DIRT because DIRT considers only slot-fillers. In contrast, BERT understands that those words are related to the concept of time in those sentences because of its attention mechanism. Therefore, we believe BIRD focused more on the "spending time" meaning of the verb spend, and this caused it to perform worse compared to DIRT on this question. In other words, while contextualized embeddings do generalize better than the lexical approach in DIRT, they also carry an enhanced risk of potential semantic drift.

Following this observation, we decided to extend Q4 to include all the meanings of the verb "spend" and evaluate BIRD as well DIRT on it. In order to do that, we added a question to the original Q4 evaluation, where we considered the meanings of "spending time" as well as "spending energy/effort" as correct. We call this question Q4e (which stands for Q4 extended). When performing the evaluation, we judge a found path as correct if it can be used to answer any of the three questions of Q4e. We performed this evaluation for Q4.2 where DIRT particularly had performed better than BIRD and called it Q4.2e. We observed that this time BIRD performed as well as DIRT in terms of correct found paths and slightly better than DIRT in terms of finding strict paraphrases.

5. Conclusions

In this work we proposed a novel implementation of Harris (1954)'s distributional hypothesis for rules. In particular, we proposed to measure the similarity between pairs of slots (i.e., the set of concepts matched by a pattern) using contextualized embeddings instead of lexical overlap. This new slot similarity measure provides a quantitative interpretation of the distributional hypothesis for rules, which states that "if two paths tend to link the same set of words, ... their meanings are similar." This, in turn, allowed us to discover new patterns that are similar to a single seed pattern in an empirical evaluation. This evaluation showed that our strategy performs considerably better than the original DIRT algorithm that inspired us.

At a higher level, this work fits in the exciting space that combines deep learning and symbolic methods. Our work carries some of the advantages of both directions: we take advantage of the better generalization power of transformer networks, but output rule-based models, which carry the interpretability of symbolic methods.

For reproducibility, we share the source code and data generated in this work at this URL: `https://github.com/clulab/releases/tree/master/lrec2022-bird`.

Acknowledgements

This work was partially supported by NSF grant #2006583, and by DARPA under the World Modelers program, grant number W911NF1810014. Mihai Surdeanu declares a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies.

6. References

Barzilay, R. and McKeown, K. R. (2001). Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pages 50–57.

Bhagat, R., Pantel, P., and Hovy, E. (2007). Ledir: An unsupervised algorithm for learning directionality of inference rules. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 161–170.

Chiticariu, L., Li, Y., and Reiss, F. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 827–832.

Chklovski, T. and Pantel, P. (2004). Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 33–40.

Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

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

Dinu, G. and Lapata, M. (2010). Topic models for meaning similarity in context. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 250–258.

Dinu, G. and Wang, R. (2009). Inference rules and their application to recognizing textual entailment. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 211–219.

Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., and Zhou, J. (2019). Fewrel 2.0: Towards more challenging few-shot relation classification. In *EMNLP/IJCNLP*.

Guo, M., Zhang, Y., Zhao, D., and Liu, T. (2019). Mining predicate-based entailment rules using deep contextual architecture. In *Neurocomputing 323*, pages 52–61.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Ibrahim, A., Katz, B., and Lin, J. (2003). Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 57–64.

Landis, J. R. and Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, pages 363–374.

Lin, D. and Pantel, P. (2001). Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Manning, C. D. (2015). Computational linguistics and deep learning. *Computational Linguistics*, 41(4):701–707.

McIntosh, T. and Curran, J. R. (2008). Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 97–105.

McIntosh, T. (2010). Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 356–365.

Melamud, O., Berant, J., Dagan, I., Goldberger, J., and Szpektor, I. (2013a). A two level model for context sensitive inference rules. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1331–1340.

Melamud, O., Dagan, I., Goldberger, J., and Szpektor, I. (2013b). Using lexical expansion to learn inference rules from sparse data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–288.

Pantel, P., Bhagat, R., Coppola, B., Chklovski, T., and Hovy, E. (2007). Isp: Learning inferential selectional preferences. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 564–571.

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual meeting of the association for Computational Linguistics*, pages 41–47.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Riloff, E., Jones, R., et al. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, pages 1044–1049.

Roberto, B., De Cao, D., Marocco, P., and Pennacchiotti, M. (2007). Learning selectional preferences for entailment or paraphrasing rules. In *RANLP*.

Sabo, O. M. S., Elazar, Y., Goldberg, Y., and Dagan, I. (2021). Revisiting few-shot relation classification: Evaluation data and classification schemes. *Transactions of the Association for Computational Linguistics*, 9:691–706.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., and Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28:2503–2511.

Shinyama, Y. and Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 304–311.

Soares, L. B., FitzGerald, N., Ling, J., and Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Sun, A. and Grishman, R. (2010). Semi-supervised semantic pattern discovery with guidance from unsupervised pattern clusters. In *Proceedings of the 23rd International Confer-*

ence on Computational Linguistics: Posters, pages 1194–1202.

Szpektor, I. and Dagan, I. (2008). Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856.

Szpektor, I., Dagan, I., Bar-Haim, R., and Goldberger, J. (2008). Contextual preferences. In *Proceedings of ACL-08: HLT*, pages 683–691.

TFDS Team. (2021). TensorFlow Datasets, a collection of ready-to-use datasets. `https://www.tensorflow.org/datasets`.

Valenzuela-Escárcega, M. A., Hahn-Powell, G., Bell, D., and Surdeanu, M. (2016). Snaptogrid: From statistical to interpretable models for biomedical information extraction. *arXiv preprint arXiv:1606.09604*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Voorhees, E. M. et al. (1999). The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82. Citeseer.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, October.

Yangarber, R. (2003). Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 343–350.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.