Matching the Blanks: Distributional Similarity for Relation Learning

Livio Baldini Soares **Nicholas FitzGerald** **Jeffrey Ling*** **Tom Kwiatkowski**
Google Research
{liviobs,nfitz,jeffreyling,tomkwiat}@google.com

Abstract

General purpose relation extractors, which can model arbitrary relations, are a core aspiration in information extraction. Efforts have been made to build general purpose extractors that represent relations with their surface forms, or which jointly embed surface forms with relations from an existing knowledge graph. However, both of these approaches are limited in their ability to generalize. In this paper, we build on extensions of Harris' distributional hypothesis to relations, as well as recent advances in learning text representations (specifically, BERT), to build task agnostic relation representations solely from entity-linked text. We show that these representations significantly outperform previous work on exemplar based relation extraction (FewRel) even without using any of that task's training data. We also show that models initialized with our task agnostic representations, and then tuned on supervised relation extraction datasets, significantly outperform the previous methods on SemEval 2010 Task 8, KBP37, and TACRED.

1 Introduction

Reading text to identify and extract relations between entities has been a long standing goal in natural language processing (Cardie, 1997). Typically efforts in relation extraction fall into one of three groups. In a first group, supervised (Kambhatla, 2004; GuoDong et al., 2005; Zeng et al., 2014), or distantly supervised relation extractors (Mintz et al., 2009) learn a mapping from text to relations in a limited schema. Forming a second group, open information extraction removes the limitations of a predefined schema by instead representing relations using their surface forms (Banko et al., 2007; Fader et al., 2011; Stanovsky et al., 2018), which increases scope but also leads to an associated lack of generality since many surface forms can express the same relation. Finally, the universal schema (Riedel et al., 2013) embraces both the diversity of text, and the concise nature of schematic relations, to build a joint representation that has been extended to arbitrary textual input (Toutanova et al., 2015), and arbitrary entity pairs (Verga and McCallum, 2016). However, like distantly supervised relation extractors, universal schema rely on large knowledge graphs (typically Freebase (Bollacker et al., 2008)) that can be aligned to text.

Building on Lin and Pantel (2001)'s extension of Harris' distributional hypothesis (Harris, 1954) to relations, as well as recent advances in learning word representations from observations of their contexts (Mikolov et al., 2013; Peters et al., 2018; Devlin et al., 2018), we propose a new method of learning relation representations directly from text. First, we study the ability of the Transformer neural network architecture (Vaswani et al., 2017) to encode relations between entity pairs, and we identify a method of representation that outperforms previous work in supervised relation extraction. Then, we present a method of training this relation representation without any supervision from a knowledge graph or human annotators by *matching the blanks*.

> [BLANK], inspired by Cale's earlier cover, recorded one of the most acclaimed versions of "[BLANK]"
>
> [BLANK]'s rendition of "[BLANK]" has been called "one of the great songs" by Time, and is included on Rolling Stone's list of "The 500 Greatest Songs of All Time".

Figure 1: "Matching the blanks" example where both relation statements share the same two entities.

Following Riedel et al. (2013), we assume access to a corpus of text in which entities have been linked to unique identifiers and we define a *rela-*

*Work done as part of the Google AI residency.

tion statement to be a block of text containing two marked entities. From this, we create training data that contains relation statements in which the entities have been replaced with a special [BLANK] symbol, as illustrated in Figure 1. Our training procedure takes in pairs of blank-containing relation statements, and has an objective that encourages relation representations to be similar if they range over the same pairs of entities. After training, we employ learned relation representations to the recently released FewRel task (Han et al., 2018) in which specific relations, such as *'original language of work'* are represented with a few exemplars, such as *The Crowd (Italian: La Folla) is a 1951 Italian film*. Han et al. (2018) presented FewRel as a supervised dataset, intended to evaluate models' ability to adapt to relations from new domains at test time. We show that through training by matching the blanks, we can outperform Han et al. (2018)'s top performance on FewRel, without having seen any of the FewRel training data. We also show that a model pre-trained by matching the blanks and tuned on FewRel outperforms humans on the FewRel evaluation. Similarly, by training by matching the blanks and then tuning on labeled data, we significantly improve performance on the SemEval 2010 Task 8 (Hendrickx et al., 2009), KBP-37 (Zhang and Wang, 2015), and TACRED (Zhang et al., 2017) relation extraction benchmarks.

2 Overview

Task definition In this paper, we focus on learning mappings from *relation statements* to *relation representations*. Formally, let $\mathbf{x} = [x_0 \dots x_n]$ be a sequence of tokens, where $x_0 = [\text{CLS}]$ and $x_n = [\text{SEP}]$ are special start and end markers. Let $\mathbf{s}_1 = (i, j)$ and $\mathbf{s}_2 = (k, l)$ be pairs of integers such that $0 < i < j - 1$, $j < k$, $k \leq l - 1$, and $l \leq n$. A relation statement is a triple $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$, where the indices in $\mathbf{s}_1$ and $\mathbf{s}_2$ delimit entity mentions in $\mathbf{x}$: the sequence $[x_i \dots x_{j-1}]$ mentions an entity, and so does the sequence $[x_k \dots x_{l-1}]$. Our goal is to learn a function $\mathbf{h}_r = f_\theta(\mathbf{r})$ that maps the relation statement to a fixed-length vector $\mathbf{h}_r \in \mathcal{R}^d$ that represents the relation expressed in $\mathbf{x}$ between the entities marked by $\mathbf{s}_1$ and $\mathbf{s}_2$.

Contributions This paper contains two main contributions. First, in Section 3.1 we investigate different architectures for the relation encoder f_θ, all built on top of the widely used Transformer sequence model (Devlin et al., 2018; Vaswani et al., 2017). We evaluate each of these architectures by applying them to a suite of relation extraction benchmarks with supervised training.

Our second, more significant, contribution—presented in Section 4—is to show that f_θ can be learned from widely available distant supervision in the form of entity linked text.

3 Architectures for Relation Learning

The primary goal of this work is to develop models that produce relation representations directly from text. Given the strong performance of recent deep transformers trained on variants of language modeling, we adopt Devlin et al. (2018)'s BERT model as the basis for our work. In this section, we explore different methods of representing relations with the Transformer model.

3.1 Relation Classification and Extraction Tasks

We evaluate the different methods of representation on a suite of supervised relation extraction benchmarks. The relation extractions tasks we use can be broadly categorized into two types: fully supervised relation extraction, and few-shot relation matching.

For the supervised tasks, the goal is to, given a relation statement $\mathbf{r}$, predict a relation type $t \in \mathcal{T}$ where $\mathcal{T}$ is a fixed dictionary of relation types and $t = 0$ typically denotes a lack of relation between the entities in the relation statement. For this type of task we evaluate on SemEval 2010 Task 8 (Hendrickx et al., 2009), KBP-37 (Zhang and Wang, 2015) and TACRED (Zhang et al., 2017). More formally,

In the case of few-shot relation matching, a set of candidate relation statements are ranked, and matched, according to a query relation statement. In this task, examples in the test and development sets typically contain relation types not present in the training set. For this type of task, we evaluate on the FewRel (Han et al., 2018) dataset. Specifically, we are given K sets of N labeled relation statements $\mathcal{S}_k = \{(\mathbf{r}_0, t_0) \dots (\mathbf{r}_N, t_N)\}$ where $t_i \in \{1 \dots K\}$ is the corresponding relation type. The goal is to predict the $t_q \in \{1 \dots K\}$ for a query relation statement $\mathbf{r}_q$.

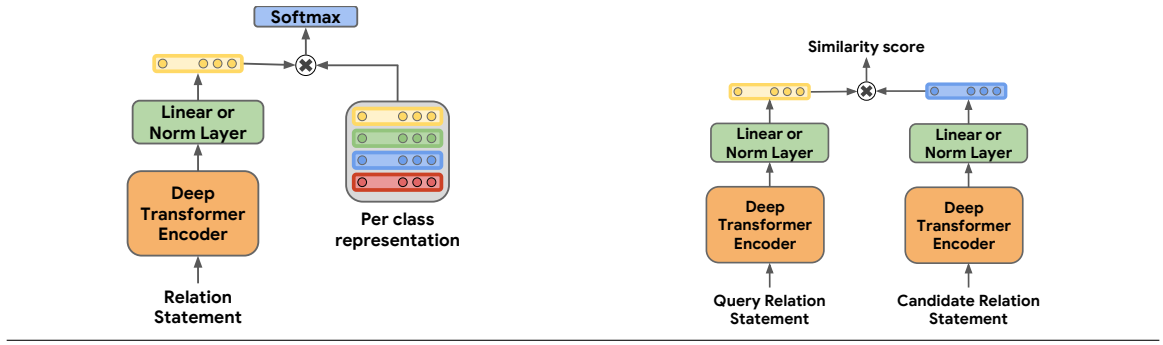
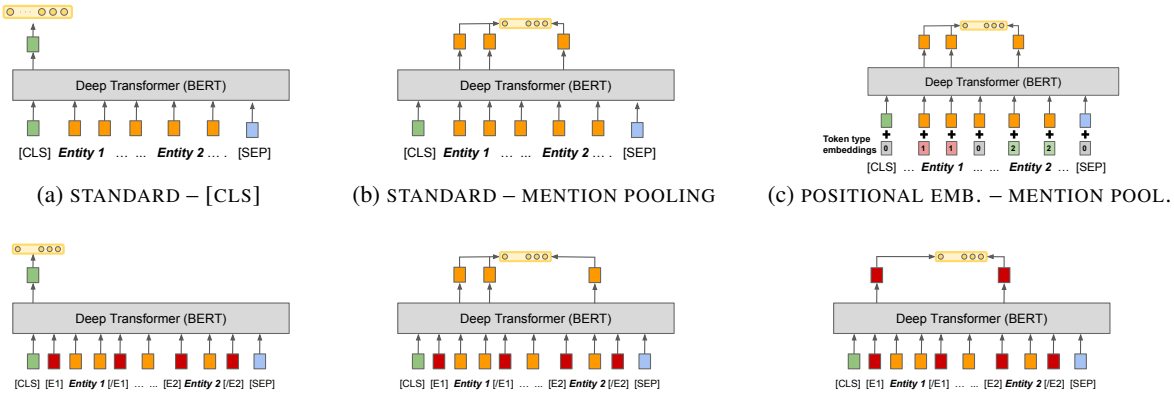


Figure 2: Illustration of losses used in our models. The left figure depicts a model suitable for supervised training, where the model is expected to classify over a predefined dictionary of relation types. The figure on the right depicts a pairwise similarity loss used for few-shot classification task.



(a) STANDARD – [CLS] (b) STANDARD – MENTION POOLING (c) POSITIONAL EMB. – MENTION POOL.

(d) ENTITY MARKERS – [CLS] (e) ENTITY MARKERS – MENTION POOL. (f) ENTITY MARKERS – ENTITY START

Figure 3: Variants of architectures for extracting relation representations from deep Transformers network. Figure (a) depicts a model with STANDARD input and [CLS] output, Figure (b) depicts a model with STANDARD input and MENTION POOLING output and Figure (c) depicts a model with POSITIONAL EMBEDDINGS input and MENTION POOLING output. Figures (d), (e), and (f) use ENTITY MARKERS input while using [CLS], MENTION POOLING, and ENTITY START output, respectively.

		SemEval 2010 Task 8		**KBP37**		**TACRED**		**FewRel 5-way-1-shot**
# training annotated examples		8,000 (6,500 for dev)		15,916		68,120		44,800
# relation types		19		37		42		100
		Dev F1	Test F1	Dev F1	Test F1	Dev F1	Test F1	Dev Acc.
Wang et al. (2016)*		–	88.0	–	–	–	–	–
Zhang and Wang (2015)*		–	79.6	–	58.8	–	–	–
Bilan and Roth (2018)*		–	84.8	–	–	–	68.2	–
Han et al. (2018)		–	–	–	–	–	–	71.6
Input type	**Output type**							
STANDARD	[CLS]	71.6	–	41.3	–	23.4	–	85.2
STANDARD	MENTION POOL.	78.8	–	48.3	–	66.7	–	87.5
POSITIONAL EMB.	MENTION POOL.	79.1	–	32.5	–	63.9	–	87.5
ENTITY MARKERS	[CLS]	81.2	–	68.7	–	65.7	–	85.2
ENTITY MARKERS	MENTION POOL.	80.4	–	68.2	–	69.5	–	87.6
ENTITY MARKERS	ENTITY START	**82.1**	**89.2**	**70**	**68.3**	**70.1**	**70.1**	**88.9**

Table 1: Results for supervised relation extraction tasks. Results on rows where the model name is marked with a * symbol are reported as published, all other numbers have been computed by us. SemEval 2010 Task 8 does not establish a default split for development; for this work we use a random slice of the training set with 1,500 examples.

3.2 Relation Representations from Deep Transformers Model

In all experiments in this section, we start with the $\text{BERT}_{\text{LARGE}}$ model made available by Devlin et al. (2018) and train towards task-specific losses. Since BERT has not previously been applied to the problem of relation representation, we aim to answer two primary modeling questions: (1) how do we represent entities of interest in the input to BERT, and (2) how do we extract a fixed length representation of a relation from BERT's output. We present three options for both the input encoding, and the output relation representation. Six combinations of these are illustrated in Figure 3.

3.2.1 Entity span identification

Recall, from Section 2, that the relation statement $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$ contains the sequence of tokens $\mathbf{x}$ and the entity span identifiers $\mathbf{s}_1$ and $\mathbf{s}_2$. We present three different options for getting information about the focus spans $\mathbf{s}_1$ and $\mathbf{s}_2$ into our BERT encoder.

Standard input First we experiment with a BERT model that does not have access to any explicit identification of the entity spans $\mathbf{s}_1$ and $\mathbf{s}_2$. We refer to this choice as the STANDARD input. This is an important reference point, since we believe that BERT has the ability to identify entities in $\mathbf{x}$, but with the STANDARD input there is no way of knowing which two entities are in focus when $\mathbf{x}$ contains more than two entity mentions.

Positional embeddings For each of the tokens in its input, BERT also adds a segmentation embedding, primarily used to add sentence segmentation information to the model. To address the STANDARD representation's lack of explicit entity identification, we introduce two new segmentation embeddings, one that is added to all tokens in the span $\mathbf{s}_1$, while the other is added to all tokens in the span $\mathbf{s}_2$. This approach is analogous to previous work where positional embeddings have been applied to relation extraction (Zhang et al., 2017; Bilan and Roth, 2018).

Entity marker tokens Finally, we augment $\mathbf{x}$ with four reserved word pieces to mark the begin and end of each entity mention in the relation statement. We introduce the $[E1_{start}]$, $[E1_{end}]$, $[E2_{start}]$ and $[E2_{end}]$ and modify $\mathbf{x}$ to give

$$\begin{aligned}\tilde{\mathbf{x}} =&[x_0 \dots [E1_{start}]\ x_i \dots x_{j-1}\ [E1_{end}] \\ &\dots [E2_{start}]\ x_k \dots x_{l-1}\ [E2_{end}] \dots x_n].\end{aligned}$$

and we feed this token sequence into BERT instead of $\mathbf{x}$. We also update the entity indices $\tilde{\mathbf{s}}_1 = (i+1, j+1)$ and $\tilde{\mathbf{s}}_2 = (k+3, l+3)$ to account for the inserted tokens. We refer to this representation of the input as ENTITY MARKERS.

3.3 Fixed length relation representation

We now introduce three separate methods of extracting a fixed length relation representation $\mathbf{h}_r$ from the BERT encoder. The three variants rely on extracting the last hidden layers of the transformer network, which we define as $H = [\mathbf{h}_0, ...\mathbf{h}_n]$ for $n = |\mathbf{x}|$ (or $|\tilde{\mathbf{x}}|$ if entity marker tokens are used).

[CLS] token Recall from Section 2 that each $\mathbf{x}$ starts with a reserved [CLS] token. BERT's output state that corresponds to this token is used by Devlin et al. (2018) as a fixed length sentence representation. We adopt the [CLS] output, $\mathbf{h}_0$, as our first relation representation.

Entity mention pooling We obtain $\mathbf{h}_r$ by max-pooling the final hidden layers corresponding to the word pieces in each entity mention, to get two vectors $\mathbf{h}_{e_1} = \text{MAXPOOL}([\mathbf{h}_i ... \mathbf{h}_{j-1}])$ and $\mathbf{h}_{e_2} = \text{MAXPOOL}([\mathbf{h}_k ... \mathbf{h}_{l-1}])$ representing the two entity mentions. We concatenate these two vectors to get the single representation $\mathbf{h}_r = \langle \mathbf{h}_{e_1} | \mathbf{h}_{e_2} \rangle$ where $\langle a|b \rangle$ is the concatenation of a and b. We refer to this architecture as MENTION POOLING.

Entity start state Finally, we propose simply representing the relation between two entities with the concatenation of the final hidden states corresponding their respective start tokens, when ENTITY MARKERS are used. Recalling that ENTITY MARKERS inserts tokens in $\mathbf{x}$, creating offsets in $\mathbf{s}_1$ and $\mathbf{s}_2$, our representation of the relation is $\mathbf{r}_h = \langle \mathbf{h}_i | \mathbf{h}_{j+2} \rangle$. We refer to this output representation as ENTITY START output. Note that this can only be applied to the ENTITY MARKERS input.

Figure 3 illustrates a few of the variants we evaluated in this section. In addition to defining the model input and output architecture, we fix the training loss used to train the models (which is illustrated in Figure 2). In all models, the output representation from the Transformer network is fed into a fully connected layer that either (1)

contains a linear activation, or (2) performs layer normalization (Ba et al., 2016) on the representation. We treat the choice of post Transfomer layer as a hyper-parameter and use the best performing layer type for each task.

For the supervised tasks, we introduce a new classification layer $\mathcal{W} \in \mathcal{R}^{KxH}$ where H is the size of the relation representation and K is the number of relation types. The classification loss is the standard cross entropy of the softmax of $h_r W^T$ with respect to the true relation type.

For the few-shot task, we use the dot product between relation representation of the query statement and each of the candidate statements as a similarity score. In this case, we also apply a cross entropy loss of the softmax of similarity scores with respect to the true class.

We perform task-specific fine-tuning of the BERT model, for all variants, with the following set of hyper-parameters:

- **Transformer Architecture**: 24 layers, 1024 hidden size, 16 heads
- **Weight Initialization**: $\text{BERT}_{\text{LARGE}}$
- **Post Transformer Layer**: Dense with linear activation (KBP-37 and TACRED), or Layer Normalization layer (SemEval 2010 and FewRel).
- **Training Epochs**: 1 to 10
- **Learning Rate (supervised)**: 3e-5 with Adam
- **Batch Size (supervised)**: 64
- **Learning Rate (few shot)**: 1e-4 with SGD
- **Batch Size (few shot)**: 256

Table 1 shows the results of model variants on the three supervised relation extraction tasks and the 5-way-1-shot variant of the few-shot relation classification task. For all four tasks, the model using the ENTITY MARKERS input representation and ENTITY START output representation achieves the best scores.

From the results, it is clear that adding positional information in the input is critical for the model to learn useful relation representations. Unlike previous work that have benefited from positional embeddings (Zhang et al., 2017; Bilan and Roth, 2018), the deep Transformers benefits the most from seeing the new entity boundary word pieces (ENTITY MARKERS). It is also worth noting that the best variant outperforms previous published models on all four tasks. For the remainder of the paper, we will use this architecture when further training and evaluating our models.

4 Learning by Matching the Blanks

So far, we have used human labeled training data to train our relation statement encoder f_θ. Inspired by open information extraction (Banko et al., 2007; Angeli et al., 2015), which derives relations directly from tagged text, we now introduce a new method of training f_θ without a predefined ontology, or relation-labeled training data. Instead, we declare that for any pair of relation statements $\mathbf{r}$ and $\mathbf{r}'$, the inner product $f_\theta(\mathbf{r})^\top f_\theta(\mathbf{r}')$ should be high if the two relation statements, $\mathbf{r}$ and $\mathbf{r}'$, express semantically similar relations. And, this inner product should be low if the two relation statements express semantically different relations.

Unlike related work in distant supervision for information extraction (Hoffmann et al., 2011; Mintz et al., 2009), we do not use relation labels at training time. Instead, we observe that there is a high degree of redundancy in web text, and each relation between an arbitrary pair of entities is likely to be stated multiple times. Subsequently, $\mathbf{r} = (\mathbf{x}, \mathbf{s}_1, \mathbf{s}_2)$ is more likely to encode the same semantic relation as $\mathbf{r}' = (\mathbf{x}', \mathbf{s}'_1, \mathbf{s}'_2)$ if $\mathbf{s}_1$ refers to the same entity as $\mathbf{s}'_1$, and $\mathbf{s}_2$ refers to the same entity as $\mathbf{s}'_2$. Starting with this observation, we introduce a new method of learning f_θ from entity linked text. We introduce this method of learning by *matching the blanks* (MTB). In Section 5 we show that MTB learns relation representations that can be used without any further tuning for relation extraction—even beating previous work that trained on human labeled data.

4.1 Learning Setup

Let $\mathcal{E}$ be a predefined set of entities. And let $\mathcal{D} = [(\mathbf{r}^0, e_1^0, e_2^0) \dots (\mathbf{r}^N, e_1^N, e_2^N)]$ be a corpus of relation statements that have been labeled with two entities $e_1^i \in \mathcal{E}$ and $e_2^i \in \mathcal{E}$. Recall, from Section 2, that $\mathbf{r}^i = (\mathbf{x}^i, \mathbf{s}_1^i, \mathbf{s}_2^i)$, where $\mathbf{s}_1^i$ and $\mathbf{s}_2^i$ delimit entity mentions in $\mathbf{x}^i$. Each item in $\mathcal{D}$ is created by pairing the relation statement $\mathbf{r}^i$ with the two entities e_1^i and e_2^i corresponding to the spans $\mathbf{s}_1^i$ and $\mathbf{s}_2^i$, respectively.

We aim to learn a relation statement encoder f_θ that we can use to determine whether or not two relation statements encode the same relation. To do this, we define the following binary classifier

$$p(l = 1 | \mathbf{r}, \mathbf{r}') = \frac{1}{1 + \exp f_\theta(\mathbf{r})^\top f_\theta(\mathbf{r}')}$$

to assign a probability to the case that $\mathbf{r}$ and $\mathbf{r}'$ encode the same relation ($l = 1$), or not ($l = 0$). We will then learn the parameterization of f_θ that

$\mathbf{r}_A$	In 1976, e_1 (then of Bell Labs) published e_2, the first of his books on programming inspired by the Unix operating system.
$\mathbf{r}_B$	The "e_2" series spread the essence of "C/Unix thinking" with makeovers for Fortran and Pascal. e_1's Ratfor was eventually put in the public domain.
$\mathbf{r}_C$	e_1 worked at Bell Labs alongside e_3 creators Ken Thompson and Dennis Ritchie.
Mentions	e_1 = Brian Kernighan, e_2 = Software Tools, e_3 = Unix

Table 2: Example of "matching the blanks" automatically generated training data. Statement pairs r_A and r_B form a positive example since they share resolution of two entities. Statement pairs r_A and r_C as well as r_B and r_C form strong negative pairs since they share one entity in common but contain other non-matching entities.

minimizes the loss

$$\mathcal{L}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|^2} \sum_{(\mathbf{r},e_1,e_2)\in\mathcal{D}} \sum_{(\mathbf{r}',e_1',e_2')\in\mathcal{D}} \delta_{e_1,e_1'}\delta_{e_2,e_2'} \cdot \log p(l=1|\mathbf{r},\mathbf{r}') + (1-\delta_{e_1,e_1'}\delta_{e_2,e_2'}) \cdot \log(1-p(l=1|\mathbf{r},\mathbf{r}')) \quad (1)$$

where $\delta_{e,e'}$ is the Kronecker delta that takes the value 1 iff $e = e'$, and 0 otherwise.

4.2 Introducing Blanks

Readers may have noticed that the loss in Equation 1 can be minimized perfectly by the entity linking system used to create $\mathcal{D}$. And, since this linking system does not have any notion of relations, it is not reasonable to assume that f_θ will somehow magically build meaningful relation representations. To avoid simply relearning the entity linking system, we introduce a modified corpus

$$\tilde{\mathcal{D}} = [(\tilde{\mathbf{r}}^0, e_1^0, e_2^0) \ldots (\tilde{\mathbf{r}}^N, e_1^N, e_2^N)]$$

where each $\tilde{\mathbf{r}}^i = (\tilde{\mathbf{x}}^i, \mathbf{s}_1^i, \mathbf{s}_2^i)$ contains a relation statement in which one or both entity mentions may have been replaced by a special [BLANK] symbol. Specifically, $\tilde{\mathbf{x}}$ contains the span defined by $\mathbf{s}_1$ with probability α. Otherwise, the span has been replaced with a single [BLANK] symbol. The same is true for $\mathbf{s}_2$. Only α^2 of the relation statements in $\tilde{D}$ explicitly name both of the entities that participate in the relation. As a result, minimizing $\mathcal{L}(\tilde{\mathcal{D}})$ requires f_θ to do more than simply identifying named entities in $\mathbf{r}$. We hypothesize that training on $\tilde{D}$ will result in a f_θ that encodes the semantic relation between the two possibly elided entity spans. Results in Section 5 support this hypothesis.

4.3 Matching the Blanks Training

To train a model with matching the blank task, we construct a training setup similar to BERT, where two losses are used concurrently: the masked language model loss and the matching the blanks loss. For generating the training corpus, we use English Wikipedia and extract text passages from the HTML paragraph blocks, ignoring lists, and tables. We use an off-the-shelf entity linking system[1] to annotate text spans with a unique knowledge base identifier (e.g., Freebase ID or Wikipedia URL). The span annotations include not only proper names, but other referential entities such as common nouns and pronouns. From this annotated corpus we extract relation statements where each statement contains at least two grounded entities within a fixed sized window of tokens[2]. To prevent a large bias towards relation statements that involve popular entities, we limit the number of relation statements that contain the same entity by randomly sampling a constant number of relation statements that contain any given entity.

We use these statements to train model parameters to minimize $\mathcal{L}(\tilde{\mathcal{D}})$ as described in the previous section. In practice, it is not possible to compare every pair of relation statements, as in Equation 1, and so we use a noise-contrastive estimation (Gutmann and Hyvärinen, 2012; Mnih and Kavukcuoglu, 2013). In this estimation, we consider all positive pairs of relation statements that contain the same entity, so there is no change to the contribution of the first term in Equation 1—where $\delta_{e_1,e_1'}\delta_{e_2,e_2'} = 1$. The approximation does, however, change the contribution of the second term.

Instead of summing over all pairs of relation statements that do not contain the same pair of entities, we sample a set of negatives that are either randomly sampled uniformly from the set of all relation statement pairs, or are sampled from the set of relation statements that share just a single

[1] We use the public Google Cloud Natural Language API to annotate our corpus extracting the "entity analysis" results — https://cloud.google.com/natural-language/docs/basics#entity_analysis .

[2] We use a window of 40 tokens, which we observed provides some coverage of long range entity relations, while avoiding a large number of co-occurring but unrelated entities.

	5-way 1-shot	5-way 5-shot	10-way 1-shot	10-way 5-shot
Proto Net	69.2	84.79	56.44	75.55
BERT$_{EM}$+MTB	**93.9**	**97.1**	**89.2**	**94.3**
Human	92.22	–	85.88	–

Table 3: Test results for FewRel few-shot relation classification task. Proto Net is the best published system from Han et al. (2018). At the time of writing, our BERT$_{EM}$+MTB model outperforms the top model on the leaderboard (http://www.zhuhao.me/fewrel/) by over 10% on the 5-way-1-shot and over 15% on the 10-way-1-shot configurations.

	SemEval 2010	KBP37	TACRED
SOTA	84.8	58.8	68.2
BERT$_{EM}$	89.2	68.3	70.1
BERT$_{EM}$+MTB	**89.5**	**69.3**	**71.5**

Table 4: F1 scores of BERT$_{EM}$+MTB and BERT$_{EM}$ based relation classifiers on the respective test sets. Details of the SOTA systems are given in Table 1.

entity. We include the second set 'hard' negatives to account for the fact that most randomly sampled relation statement pairs are very unlikely to be even remotely topically related, and we would like to ensure that the training procedure sees pairs of relation statements that refer to similar, but different, relations. Finally, we probabilistically replace each entity's mention with [BLANK] symbols, with a probability of $\alpha = 0.7$, as described in Section 3.2, to ensure that the model is not confounded by the absence of [BLANK] symbols in the evaluation tasks. In total, we generate 600 million relation statement pairs from English Wikipedia, roughly split between 50% positive and 50% strong negative pairs.

5 Experimental Evaluation

In this section, we evaluate the impact of training by matching the blanks. We start with the best BERT based model from Section 3.3, which we call BERT$_{EM}$, and we compare this to a variant that is trained with the matching the blanks task (BERT$_{EM}$+MTB). We train the BERT$_{EM}$+MTB model by initializing the Transformer weights to the weights from BERT$_{LARGE}$ and use the following parameters:

- **Learning rate**: 3e-5 with Adam
- **Batch size**: 2,048
- **Number of steps**: 1 million
- **Relation representation**: ENTITY MARKER

We report results on all of the tasks from Section 3.1, using the same task-specific training methodology for both BERT$_{EM}$ and BERT$_{EM}$+MTB.

5.1 Few-shot Relation Matching

First, we investigate the ability of BERT$_{EM}$+MTB to solve the FewRel task without any task-specific training data. Since FewRel is an exemplar-based approach, we can just rank each candidate relation statement according to its representation's inner product with the exemplars' representations.

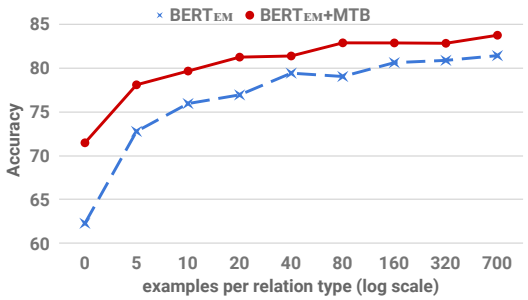
Figure 4 shows that the task agnostic BERT$_{EM}$ and BERT$_{EM}$+MTB models outperform the previous published state of the art on FewRel task even when they have not seen any FewRel training data. For BERT$_{EM}$+MTB, the increase over Han et al. (2018)'s supervised approach is very significant—8.8% on the 5-way-1-shot task and 12.7% on the 10-way-1-shot task. BERT$_{EM}$+MTB also significantly outperforms BERT$_{EM}$ in this unsupervised setting, which is to be expected since there is no relation-specific loss during BERT$_{EM}$'s training.

To investigate the impact of supervision on BERT$_{EM}$ and BERT$_{EM}$+MTB, we introduce increasing amounts of FewRel's training data. Figure 4 shows the increase in performance as we either increase the number of training examples for each relation type, or we increase the number of relation types in the training data. When given access to all of the training data, BERT$_{EM}$ approaches BERT$_{EM}$+MTB's performance. However, when we keep all relation types during training, and vary the number of types per example, BERT$_{EM}$+MTB only needs 6% of the training data to match the performance of a BERT$_{EM}$ model trained on all of the training data. We observe that maintaining a diversity of relation types, and reducing the number of examples per type, is the most effective way to reduce annotation effort for this task. The results in Figure 4 show that MTB training could be used to significantly reduce effort in implementing an exemplar based relation extraction system.

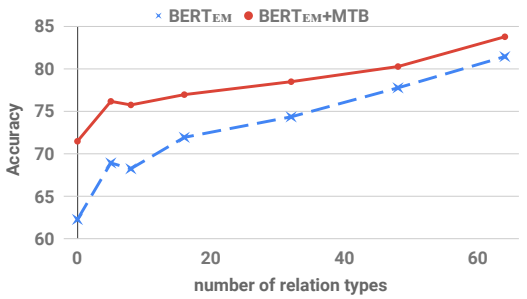
Finally, we report BERT$_{EM}$+MTB's performance on all of FewRel's fully supervised tasks in Table 3. We see that it outperforms the human upper bound reported by Han et al. (2018), and it significantly outperforms all other submissions to the FewRel leaderboard, published or unpublished.

5.2 Supervised Relation Extraction

Table 4 contains results for our classifiers tuned on supervised relation extraction data. As was established in Section 3.2, our BERT$_{EM}$ based classifiers



5 way 1 shot						
# examples per type	0	5	20	80	320	700
Prot.Net. (CNN)	–	–	–	–	–	71.6
BERTEM	72.9	81.6	85.1	86.9	88.8	88.9
BERTEM+MTB	80.4	85.5	88.4	89.6	89.6	90.1
10 way 1 shot						
# examples per type	0	5	20	80	320	700
Prot.Net. (CNN)	–	–	–	–	–	58.8
BERTEM	62.3	72.8	76.9	79.0	81.4	82.8
BERTEM+MTB	71.5	78.1	81.2	82.9	83.7	83.4



5 way 1 shot					
# training types	0	5	16	32	64
Prot.Net. (CNN)	–	–	–	–	71.6
BERTEM	72.9	78.4	81.2	83.4	88.9
BERTEM+MTB	80.4	84.04	85.5	86.8	90.1
10 way 1 shot					
# training types	0	5	16	32	64
Prot.Net. (CNN)	–	–	–	–	58.8
BERTEM	62.3	68.9	71.9	74.3	81.4
BERTEM+MTB	71.5	76.2	76.9	78.5	83.7

Figure 4: Comparison of classifiers tuned on FewRel. Results are for the development set while varying the amount of annotated examples available for fine-tuning. On the left, we display accuracies while varying the number of examples per relation type, while maintaining all 64 relations available for training. On the right, we display accuracy on the development set of the two models while varying the total number of relation types available for tuning, while maintaining all 700 examples per relation type. In both graphs, results for the 10-way-1-shot variant of the task are displayed.

% of training set	1%	10%	20%	50%	100%
SemEval 2010 Task 8					
BERTEM	28.6	66.9	75.5	80.3	82.1
BERTEM+MTB	31.2	70.8	76.2	80.4	82.7
KBP-37					
BERTEM	40.1	63.6	65.4	67.8	69.5
BERTEM+MTB	44.2	66.3	67.2	68.8	70.3
TACRED					
BERTEM	32.8	59.6	65.6	69.0	70.1
BERTEM+MTB	43.4	64.8	67.2	69.9	70.6

Table 5: F1 scores on development sets for supervised relation extraction tasks while varying the amount of tuning data available to our BERTEM and BERTEM+MTB models.

outperform previously published results for these three tasks. The additional MTB based training further increases F1 scores for all tasks.

We also analyzed the performance of our two models while reducing the amount of supervised task specific tuning data. The results displayed in Table 5 show the development set performance when tuning on a random subset of the task specific training data. For all tasks, we see that MTB based training is even more effective for low-resource cases, where there is a larger gap in performance between our BERTEM and BERTEM+MTB based classifiers. This further supports our argument that training by matching the blanks can significantly reduce the amount of human input required to create relation extractors, and populate a knowledge base.

6 Conclusion and Future Work

In this paper we study the problem of producing useful relation representations directly from text. We describe a novel training setup, which we call *matching the blanks*, which relies solely on entity resolution annotations. When coupled with a new architecture for fine-tuning relation representations in BERT, our models achieves state-of-the-art results on three relation extraction tasks, and outperforms human accuracy on few-shot relation matching. In addition, we show how the new model is particularly effective in low-resource regimes, and we argue that it could significantly reduce the amount of human effort required to create relation extractors.

In future work, we plan to work on *relation discovery* by clustering relation statements that have similar representations according to BERTEM+MTB. This would take us some of the way toward our goal of truly general purpose relation identification and extraction. We will also study representations of relations and entities that can be used to store relation triples in a distributed knowledge base. This is inspired by recent work in knowledge base embedding (Bordes et al., 2013; Nickel et al., 2016).

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 344–354.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ivan Bilan and Benjamin Roth. 2018. Position-aware self-attention with relative positional encodings for slot filling. *CoRR*, abs/1807.03052.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. Curran Associates, Inc.

Claire Cardie. 1997. Empirical methods in information extraction. *AI Magazine*, 18(4):65–80.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics.

Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809.

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

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. DIRT: Discovery of Inference Rules from Text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 323–328, New York, NY, USA. ACM Press.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*, pages 2265–2273.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 1955–1961. AAAI Press.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Patrick Verga and Andrew McCallum. 2016. Row-less universal schema. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 63–68, San Diego, CA. Association for Computational Linguistics.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1298–1307. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *CoRR*, abs/1508.01006.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.