Supervised Relation Classification as Two-way Span-Prediction

Anonymous ACL submission

Abstract

Most of the current supervised relation classification (RC) algorithms use a single embedding to represent the relation between a pair of entities. We argue that a better approach is to treat the RC task as a Span-Prediction (SP) problem, similar to Question Answering (QA). We present an SP-based system for RC and evaluate its performance compared to the embedding-based system. We demonstrate that by adding a few improvements, the supervised SP objective works significantly better than the standard classification-based objective. We achieve state-of-the-art results on the TACRED, SemEval task 8, and the CRE datasets.

1 Introduction

The relation extraction (RE) task revolves around binary relations (such as "[e1] founded [e2]"), that hold between two entities. The task is, given a corpus and a list of semantic relations, to return entity pairs e1, e2 that are connected by one of the predefined relations. This is often posed as a Relation Classification task (RC), in which we are given a sentence and two entities (where each entity is a span over the sentence), and need to classify the relation into one of |R| possible relations or to a null "no-relation" class if none of the |R| relations hold between the given entities. RE datasets, including the popular and large TACRED dataset (Zhang et al., 2017), all take the relation classification view, by providing tuples of the form (s,e1,e2,r), where s is a sentence, e1, e2 are entities in s and r is a semantic relation between e1 and e2. Consequently, the state-of-the-art models follow the classification view: the sentence and entities are encoded into a vector representation, which is then being classified into one of the R relations. The training objective then aims to embed the sentence + entities into a space in which the different relations are well separated. We argue that this is a sub-optimal training architecture and training objective for the task, and propose to use span-predictions (SP) models, as used in question-answering models, as an alternative. Our method can be summarised as follows: We start by converting each sample in the RC datasets into several new SP subsamples, where each of the subsamples is added with a predefined semantic indicator that represents a specific relation (e.g. "When was X born?" or "per::date_of_birth X"). Using the new subsamples, we train a dedicated SP model. For inference, we split each test sample into subsamples as before, evaluate each of them independently and aggregate the result to return a prediction for the entities relation type. We show a high-level flow of our method in Figure 1. Surprisingly, even when the model is exposed to tens of thousands of samples during training, the added semantic information in the templates gives a significant boost to the model, increasing its accuracy by 1.8F1.

Alt et al. (2020) analyzed the errors of current RC systems and estimated that most errors originate from incorrectly predicting "no relation" (approx. 63.5%) and by considering wrong arguments in the input sentence (approx. 10.7%). Our model is specially tailored to minimize these errors. We demonstrate this on the TACRED and SemEval datasets. Our method surpasses the current state-of-the-art on these datasets by 2.3F1 points on TACRED and 0.9F1 points on SemEval. Additionally, we experiment with the newly released “challenge relation extraction” (CRE) dataset, which was made specifically to test the existence of shallow heuristics in RE models. We surpass current state-of-the-art models by 5.0F1. On all three datasets, our span-prediction models outperform existing RC methods.

To summarize our contribution, while all current approaches to supervised relation classification use an embedding-based technique, we propose a span-prediction-based one, which significantly...
improves state-of-the-art scores on several well-known datasets.

2 Related Work

The framework of question answering was used to solve a variety of NLP tasks like coreference resolution (Wu et al., 2019), event extractions (Du and Cardie, 2020), nested named entities (Li et al., 2019a), multi-turn entity extraction (Li et al., 2019b), and many more (Jiang et al., 2019).

Levy et al. (2017) suggested using QA for zero-shot RE/RC by framing each relation by a predefined question. Our work can be seen as the fully supervised variation of his work. While the supervised task is easier in the sense that there is a large dataset available for training, it poses a challenge in the level of accuracy the system is expected to provide.

Another key work in this area is the work of Wei et al. (2019), who proposed to use span prediction method to improve RE for a corpus with more than one relation in the same sentence. They do this by identifying head entity spans, and then using SP for predicting the tail for each span, along with its relation. Our work focuses on RC instead of RE, and is restricted to answering “yes/no” judgments about given relation1,entity1,entity2 tuples. While the methods cannot be compared directly, we did attempt to evaluate Wei et al’s method on an RC dataset, by employing it as an RE model and looking for how many of the relations in the RC datasets were recovered. This yielded very low recall scores of 0.24 on TACRED and 0.71 on semEval\(^1\). While the mentioned works used SP models to improve performance on a specific task, it is worth mentioning that other works have used QA for different reasons, like (He et al., 2015) that used QA as an easier way to annotate data for the SRL task.

3 Embedding Classification vs Span-Prediction

Embedding-Based Relation Classification

A RE sample takes the form \((c, e_1, e_2, r)\) where \(c = [c_0, \ldots, c_n]\) is a context (usually a sentence), \(e_1\) and \(e_2\) are spans that correspond to head and tail entities and are given as spans over the sentence, and \(r \in R\cup\{\emptyset\}\) is a relation from a predefined set of relations \(R\), or \(\emptyset\) indicating that no relation from \(R\) holds. RC classifier takes the form of a multi-class classifier:

\[
f_{rc}(c, e_1, e_2) \mapsto r \in R \cup \{\emptyset\}
\]

The training objective is to score the correct \(r \in R \cup \{\emptyset\}\) over all other incorrect answers, usually using a cross-entropy loss. State-of-the-art methods (Baldini Soares et al., 2019) achieve this by learning an embedding function \(\text{embed}(c, e_1, e_2)\) that maps instances with the same relation to be close to the embedding of the corresponding relation in an embedding space. The embedding function is used on pre-trained masked LMs such as SpanBERT (Joshi et al., 2020), RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019).

Span Prediction

A SP sample takes the form of \((c, q, e_a)\) where \(c = [c_0, \ldots, c_n]\) is a context (a sentence or a paragraph), \(q = [q_0, \ldots, q_l]\) is a query, and \(e_a\) is the answer to the query represented as a span over \(c\), or a special out-of-sequence-span indicating that the answer does not exist.\(^2\) SP model takes the form of a span predictor from a \(c, q\) pair to a span over \(c\):

\[
f_{qa}(c, q) \mapsto e_a \in [1..m] \times [1..m]
\]

This predictor takes the form:

\[
\arg\max_{e_a} \text{score}_{c,q}(e_a)
\]

where \(\text{score}_{c,q}(e_a)\) is a learned span scoring function, and \(e_a\) ranges over all possible spans. The training objective is to maximize the score of the correct spans above all other candidate spans. The scoring function in state-of-the-art models (McCann et al., 2018; He et al., 2015; Wu et al., 2019) also make use of pre-trained LMs.

3.1 Method Comparison

The question \(q\) (“where was Sam born?”) in QA can be thought of as involving a span \(e_q\) (“Sam”) and a predicate \(r_q\) (“where was born”). Under this view, the SP classifier can be written as:

\[
f_{qa}(c, e_q, r_q) \mapsto e
\]

compared to the relation classifier:

\[
f_{rc}(c, e_1, e_2) \mapsto r
\]

\(^1\)We tried to make the tasks more similar by skipping the head-span prediction step and feeding the algorithm with the gold head entity in the sentence, and letting it infer the relevant relations and their tails. This yielded also similarly low recall.

\(^2\)In practice, this span is the out-of-sentence CLS token.
Figure 1: Traditional RC (top) VS our span-prediction approach (bottom). For each relation type that is compatible with the marked entity type, we create two questions. If the model answers one of them correctly, we assert the relation over the two entities.

Note that both methods include a context, two spans, and a relation/predicate, but the RC models classify from two spans to a relation (from a fixed set), while the SP model classify from a span and a relation (from a potentially open set) into another span.

Let’s review the implications of this difference:

**Embedding** While both the traditional and SP methods embed the input prior to classification, the items that are being embedded in each, change. In traditional RC the embedding $h_{re}$ is based on the context and entities:

$$h_{re} = \text{embed}(c, e_1, e_2)$$

while for SP RC the embedding $h_{qa}$ encodes both the context and the question (the relation of interest and one of the entities):

$$h_{qa} = \text{embed}(q, c) = \text{embed}(r, e_1, c)$$

Note that the SP embedding includes the relation name, as well as template words that surround the $(r, e_1)$ pair. This has several benefits, as we explain below.

---

3.2 Implications

**Relation type indication for the pretrained model.** The inclusion of the relation $r$ in the input to the contextualized embedder allows the embedder to specialize on a specific relation. For example, consider the sentence “Martha gave birth to John last February”. The entity John participates in two relations: “date of birth” and “parents of”. The RC embedding will have to either infer the relation based on the entities, or else preserve information regarding both relations, while in the span-prediction case the embedding takes the relation $r$ into account, and can focus on the existence (or nonexistence) of one of the entities as the argument for this relation. Focusing on a specific relation in the embedding stage (which involves most of the model computation) allows using all of the model computation for a specific relation.

**Sharing of semantic information.** The span-prediction model is based on templates encoding $r$ and $e$, and these templates may pass valuable information to the model: (1) by containing semantic information that is correlated to the target relation (e.g. questions that represent the relation); and (2) coding the full sentence and adding the focus for each relation in the final layer of classification (as used by Wei et al. (2019)): $h_c = \text{embed}(c)$ While this representation considers less information than the previous one (resulting in lower accuracy), it can be used to classify multiple entities and relation with one calculation, which make it more computationally efficient.
by containing information that can help generalize over different relations.

For example, consider the relation “born in” with the template question “Who was born in X?” and the relation “parent of” with the question “Who is the parent of X?”. While the relations are different from each other, they both contain an entity of type “person”, a similarity which is communicated to the model by the use of the shared word “Who”. This can help the model generalize commonalities across relation types when needed. While the template input might look insignificant in the supervision setting, where training data is abundant, in practice it has a significant effect on the overall model performance, as shown in Section 5.2.

More demanding loss function. During training, relation-classification models classify sentences with marked entities to one of |R| + 1 relation types. Span prediction models are also required to decide whether the sentence contains a given relation (they should predict if the sentence contains the answer or not), but they are also required to predict the span of the missing argument. This means that the span-prediction models are required to predict the relation between the input entities in addition to the relation itself.

Limitations. It is important to note, however, that the span-prediction method is more computationally expensive: instead of performing a single contextualized embedding operation followed by k + 1-way classification, we need to perform k contextualized embedding operations (and in our case, 2k such operations), each of them followed by the scoring of all spans. We leave ways of improving the computational efficiency of the model to future work.

4 Reducing RC to span-prediction

Given the uncovered similarity between RC and span-predicting showed in Section 3.1, we now describe how to reduce RC to SP.

Given an RC instance \((c, e_1, e_2) \rightarrow rel\) we can create an SP instance \((q = (e_q, rel_q), e) \rightarrow e_a\) as follows. Let \(T_{rel}(e)\) be a template function associated with relation \(rel\). The function takes an entity \(e\) and returns a question. For example, a template for date-of-birth relation might be \(T_{dob} = \text{“When was ___ born?”}\), and \(T_{dob}(\text{Sam}) = \text{“When was Sam born?”}\). Given an RC instance \((c, e_1, e_2) \rightarrow rel\) we can now create a span-prediction instance \((T_{rel}(e_1), c) \rightarrow e_2\), and return that the relation \(rel\) holds if the span returned from \(f_{q}(T_{rel}(e_1), c)\) is compatible with \(e_2\). This is essentially the construction of Levy et al. (2017). We extend it as follows:

Bidirectional questions. We note that the decision to predict \(e_2\) based on \(e_1\) is arbitrary, and that the opposite direction can also be used using the template “Who was born on ___?” to predict \(e_1\) from \(e_2\).

We propose to use both options, by associating a relation \(rel\) with two templates, \(T_{rel} \rightarrow e_2\) and \(T_{rel} \rightarrow e_1\), creating the two corresponding SP instances, and combining the two answers. Concretely, given the RC instance:

\[ RC: (c, \text{Sam}, 1991) \rightarrow \text{date-of-birth} \]

we create the two SP instances:

\[ QA1: (c, \text{When was Sam born?}) \rightarrow 1991 \]
\[ QA2: (c, \text{Who was born in 1991?}) \rightarrow \text{Sam} \]

We show in Section 5 that using two questions indeed results in substantial improvements.

Template formulation. Note that while in this example we formulate the questions in English, a simpler template might also be used. We also experiment with a template that replaces the question by the relation name and another template that used an unused token for each relation. We elaborate on the template variations in detail in Section 5.

Answer combination. There are various possible strategies to combining the two answers. An approach which we found to be effective is to combine using an \(OR\) operation: if either of the returned spans is compatible with the expected span, the relation \(rel\) is returned, and if neither of them is compatible, the answer is no-relation.

A natural alternative is to combine using an \(AND\) operation, requiring the answers of the two questions to be compatible in order to return \(rel\). In our experiments (Section 7), this yielded lower F-scores on the relation classification task, as we classified more cases as no-relation when we shouldn’t have. The span predictor network had an easier time answering one formulation on some instances, and the other formulation on others. As span-prediction model quality improves, future applications may reconsider the combination method.

\(^5\)Two non-empty spans are said to be compatible if either of them contains the other.
### Binary vs. Multiclass.

The above reduction targets a binary version of RC, where the relation is given and the classifier needs to decide if it holds or not. We extend it to the multi-class version by creating a version for each of the relevant relations.\(^7\)

### Supervised dataset construction.

The reduction allows us to train a SP model to classify RC instances. For each RC training instance \((c, e_1, e_2, r)\), where \(r \in R \cup \{\emptyset\}\), we consider all relations \(r' \in R\) which are compatible with \((e_1, e_2)\).\(^8\) We then generate two SP instances for each of the compatible relations. Instances that are generated with the templates of the gold-relation \(r\) are marked as positive instances (their answer is either \(e_1\) or \(e_2\), as appropriate), while instances that are generated from \(r' \neq r\) are negative examples (their answer is the no-answer span). Figure 2 provides an example.

### Per-template thresholds.

General purpose SP models use a global threshold \(\tau\) to distinguish between answerable and non-answerable questions given a context. The model output given input sample \(s\) is:

\[
\text{pred}(s) = \begin{cases} 
  e & \text{if } \text{score}(e) - \text{score}(\text{no-answer}) > \tau \\
  "\text{NA}" & \text{else}
\end{cases}
\]

Where \(e\) is the token with the highest score and no-answer is the no answer span. In the supervised relation classification case, the set of questions is fixed in advance to \(2|R|\). We observe that the optimal threshold value for each question is different. We thus set a different threshold value \(\tau_{rel}\) for each template. The threshold is set by converting each labelled sample \(s\) into the tuple \((v_s, a_s)\), where \(s_v = \text{score}(e_s) - \text{score}(\text{no-answer})\) and \(a_s\) is a label that equal to 1 iff the sample has an answer or not. To find a relation specific threshold \(\tau_r\) use the following equation:

\[
\text{best}_{\tau_r}(D_r) = \arg\max_{\tau} \sum_{s \in D_r} \text{threshold}(s, \tau).
\]

Where \(D_r\) is a dataset subset that contain subsamples with a template that is based on relation \(r\)

\[
\text{threshold}(s, \tau) = \begin{cases} 
  a_s & v_s \geq \tau \\
  1 - a_s & \text{else}
\end{cases}
\]

### 5 Main Experiments

#### 5.1 Datasets

We compare ourselves on three RC datasets.

**TACRED** (Zhang et al., 2017) is currently the most popular and largest RC dataset. It spans 41 “classic” RC relations, which hold between persons, locations, organizations, dates, and so on (e.g., "siblings", "dates of birth", "subsidiaries", etc). TACRED contains 106,264 labeled sentences (train + dev + test), where 20% of the data is composed from the 41 relations and the rest 80% are “no relation” instances.

**SemEval 2010 Task 8** (SemEval, Hendrickx et al. (2010)), is a smaller dataset, containing 10,717 annotated examples covering 9 relations, without no-relation examples. SemEval relations...
are substantially different from those in TACRED, covering more abstract relations such as part-whole, cause-effect, content-container, and so on.

**Challenge relation extraction (CRE)** Rosenman et al. (2020) showed that current RC models have a strong bias towards shallow heuristics that do not capture the deep semantic relation between entities. For example, classifying an entity pair by the entities type + an unrelated event in the sentence. To show this bias empirically, they created a Wikipedia-based dataset intended to be used only for testing, which contains 3000 manually tagged sentences from the TACRED relations. Each sentence in the dataset contains two entity pairs that are compatible with the same relation. The evaluation of the CRE is binary — the model goal is to indicate if a given relation is found or not found in the dataset. The model was evaluated with both SP and RC models.

### 5.2 Template variations.

We convert the TACRED and SemEval training sets to span-prediction form in three ways, representing various amounts of semantic information. From most informative to least, the variations are:

- **Natural language questions** (question) For each RC sample we create two samples, as described in Section 4. The complete template list is available in the supplementary materials.

- **Relation name** (relation) Same as the question dataset, but we replace each of the questions with the relation name, entity, and a marker that indicate if it’s a head or tail entity. E.g., the relation
  
  \[
  \text{RC:}(c, \text{John}, \text{CEO}) \rightarrow \text{per:title}
  \]

  will be represented as the questions:

  - QA1: \((c, \text{per:title t John}) \rightarrow \text{CEO}\)
  - QA2: \((c, \text{per:title h CEO}) \rightarrow \text{John}\)

- **Unique tokens** (token) Same as the relation dataset, but we replace the relation name with a new reserved token. E.g., the above \textit{per:title} relation will be represented as the questions:

  - QA1: \((c, \text{r2 t John}) \rightarrow \text{CEO}\)
  - QA2: \((c, \text{r2 h CEO}) \rightarrow \text{John}\)

  Each of the datasets used the same train/validation/test splits.

### 5.3 Comparisons

We compare our results to several leading models, reporting the results from the corresponding papers.

- **MTB** (Baldini Soares et al., 2019) is a state of the art RC model which is based on BERT-large, and which does not involve any additional training material except for the pre-trained LM. MTB’s way of creating sentence embedding is the current state-of-the-art, and thus our most direct comparison.

- **KEPLER** (Wang et al., 2019) This model holds the current highest reported RC results over TACRED. It is a RoBERTa based RC model which incorporates additional knowledge in the form of a knowledge graph derived from Wikipedia and Wikidata and uses MTB for sentence embedding.

- **LiTian** (Li and Tian, 2020) is the current top-scoring model on the SemEval dataset. It uses a dedicated RC architecture and uses the BERT pre-trained LM.

  We train span-predicting models using the architecture described in (Devlin et al., 2018), starting from either the BERT-Large (Devlin et al., 2018) or ALBERT (Lan et al., 2019) pre-trained LMs.

  BERT-large is used to compare the state-of-the-art model reported in (Baldini Soares et al., 2019) on equal grounds, while ALBERT is a stronger pre-trained LM which is used to show the full capabilities of our approach.

### 6 Main Results

The results of the CRE evaluation are presented in Table 1. We report the results in the same format used in the original paper: the percentage of positive samples that were identified correctly (Acc+), the percentage of negative samples that were identified correctly (Acc−), and the overall weighted accuracy (Acc). Except for the token-BERT reduction, all of the reductions we used surpassed their RC and SQuAD trained models, where the SP model (BERT and ALBERT) improve by more than 5% compared to the squad models. We also

---

9The same paper reports additional results based on external training data, which is not comparable. However, these results have since been superseded by the KEPLER model.

10We used the implementations provided by Huggingface (Wolf et al., 2019). Following previous work, used the Adam optimizer, an initial learning rate of \(3e^{-5}\), and up to 20,000 steps with early stopping on a dev-set.

11We also ran preliminary tests using (Liu et al., 2019) and (Joshi et al., 2020) that showed inferior results compared to ALBERT.
We explain this anomaly by looking at the relation names in TACRED, the current best performing model on this dataset. All the supervised SP-ALBERT models outperform KEPLER. Bottom: Ablations. We show that using one-way questions (SP\textsubscript{question, ALBERT}) and the AND operator (SP-AND) perform worse than two-way questions and the OR operator.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc\textsubscript{+}</th>
<th>Acc\textsubscript{-}</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC\textsubscript{BERT}</td>
<td>70.0</td>
<td>64.8</td>
<td>67.1</td>
</tr>
<tr>
<td>SQuAD\textsubscript{BERT}</td>
<td>62.9</td>
<td>70.9</td>
<td>67.4</td>
</tr>
<tr>
<td>SP\textsubscript{token, BERT}</td>
<td>55.0</td>
<td>75.5</td>
<td>66.4</td>
</tr>
<tr>
<td>SP\textsubscript{relation, BERT}</td>
<td>66.6</td>
<td>72.1</td>
<td>69.6</td>
</tr>
<tr>
<td>SP\textsubscript{question, BERT}</td>
<td>72.5</td>
<td>75.0</td>
<td>73.9</td>
</tr>
<tr>
<td>SQuAD\textsubscript{ALBERT}</td>
<td>71.5</td>
<td>78.8</td>
<td>75.3</td>
</tr>
<tr>
<td>SP\textsubscript{token, ALBERT}</td>
<td>80.9</td>
<td>73.2</td>
<td>76.6</td>
</tr>
<tr>
<td>SP\textsubscript{relation, ALBERT}</td>
<td>78.2</td>
<td>79.8</td>
<td>79.1</td>
</tr>
<tr>
<td>SP\textsubscript{question, ALBERT}</td>
<td>81.2</td>
<td>79.5</td>
<td>80.3</td>
</tr>
</tbody>
</table>

Table 1: CRE. Span prediction model results on CRE, compared to traditional RC and QA model. RC models are relation classification models and SQuAD models are QA models that were trained on the SQuAD 2.0 dataset.

The results for SemEval are presented in Table 3. The best performing model is the QA model, which also surpasses LiTian's model. Surprisingly, the token model performs better than the relation token. We explain this anomaly by looking at the relation names in SemEval. In contrast to TACRED (and CRE) the relation names in SemEval are somewhat abstract and have lower semantic similarity to the relation instances. For example, the TACRED relation "per:parents" provides more generalization and more semantic similarity to the words that actually appear in the context compared to the SemEval relation "instrument-agency" as explained in Section 3.2.

Another difference between the datasets is the difference in accuracy gain from our models. While CRE has the most benefit, followed by TACRED and finally SemEval. We assume that this difference originates from the dataset nature — The "shallow heuristics" shown by (Rosenman et al., 2020) that CRE was made to highlight are more prominent in TACRED than SemEval. Our span-prediction based loss is specially tailored to deal with such situations. In contrast, SemEval does not contain the "no relation" type, and the chance of any two relations appearing in the same sentence is low, resulting in this challenge being a lot less prominent on SemEval than TACRED.

Since we didn't have access to KEPLER (the current state-of-the-art), we used the best-pretrained model available to us — ALBERT model. all of our ALBERT-based relation reduction methods outperform the current best TACRED model (KEPLER) by 2.3% F\textsubscript{1}, despite KEPLER using external data.

Another anomaly is that on ALBERT, the QA reduction performed worse than the relation reduction and even the token reduction. We explain this by looking at the relation names in TACRED, which contain parts that add generalization over different relations, while also containing parts that have a strong semantic connection to the relations. E.g. per:age relation having the first part supporting generalization while the latter supports the semantic connection to the relation.

7 Ablations

The importance of bidirectional questions. To assess the impact of using questions in both directions, we also report the ALBERT-based QA-
We advocated a fully supervised training of RC models as span-prediction. How well does this compare to using existing QA models, like SQuAD, in a zero-shot setting? And can we leverage the existing knowledge in QA datasets, via pre-training? We explore these two options and conclude that while the zero-shot accuracy is impressively high, the unification of SQuAD and TACRED harms the overall accuracy.

### Zero-shot SQuAD
In light of the success of SQuAD trained model on CRE (as demonstrated by Rosenman et al. (2020)), we evaluate the SQuAD 2.0 trained model performance on TACRED, using our bidirectional reduction. In this zero-shot setup, we take a SQuAD trained model (without any modifications) and apply our reduction to evaluate the test set of TACRED.

### Joined training with SQuAD
We now attempt to leverage the SQuAD 2.0 data to improve our RC model. We train our SP\textsubscript{question,BERT} model by combining SQuAD 2.0 samples and the TACRED-SP generated questions. We do this in two ways: in the unified version we combine the two datasets simply by shuffling together the TACRED and SQuAD questions into a single dataset. In the serial version we first train on the SQuAD data and then continue training the model on TACRED data.

### Results
Table 4 lists the results. Unsurprisingly, the zero shots F\textsubscript{1} score on TACRED is substantially lower than all the supervised variants. However, the recall of the zero-shot setup is substantially higher: the SQuAD 2.0 model is very permissive. Interestingly, the additional SQuAD questions did not improve—and even substantially hurt—the SP method compared to train on only the TACRED-generated questions. This goes to highlight that the main benefits of the SP method originate from the combination of the supervised training and the span-prediction objective, and not merely from the QA form, or from the additional semantic information that is potentially embedded in the QA models.

### Conclusion
In this work, we argue for the use of span-prediction methods, typically used for QA, to replace the standard RC architectures. Our approach reduces each RC sample to a series of binary span-prediction tasks. We show that This approach achieves state-of-the-art performance in supervised settings, with the moderate cost of supplying question templates that describe the relation.

---

**Table 3:** SemEval. Supervised results on the SemEval datasets. LiTian is the current state of the art.

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F\textsubscript{1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC\textsubscript{MTB,BERT}</td>
<td>-</td>
<td>-</td>
<td>89.2</td>
</tr>
<tr>
<td>LiTian (current best)</td>
<td><strong>94.2</strong></td>
<td>88.0</td>
<td><strong>91.0</strong></td>
</tr>
<tr>
<td>SP\textsubscript{token,BERT}</td>
<td>92.8</td>
<td>88.8</td>
<td>90.7</td>
</tr>
<tr>
<td>SP\textsubscript{relation,BERT}</td>
<td>91.9</td>
<td>83.1</td>
<td>87.1</td>
</tr>
<tr>
<td>SP\textsubscript{question,BERT}</td>
<td>90.7</td>
<td><strong>93.2</strong></td>
<td><strong>91.9</strong></td>
</tr>
</tbody>
</table>

**Table 4:** Using SQuAD 2.0 (Top): Evaluating SQuAD 2.0 QA model on TACRED in a zero-shot setup, using our bidirectional SP reduction. Mid: “Fine-tuning” the SP\textsubscript{question,BERT} models on TACRED after SQuAD 2.0 pre-training. Bottom: The SP model trained without pre-training, significantly outperforming the pre-trained variants.

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F\textsubscript{1}</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQuAD 2.0 (Zero shot)</td>
<td>49.7</td>
<td><strong>78.9</strong></td>
<td>57.1</td>
</tr>
<tr>
<td>SP+Pretrain (BERT,unified)</td>
<td>68.3</td>
<td>63.2</td>
<td>65.5</td>
</tr>
<tr>
<td>SP+Pretrain (BERT,serial)</td>
<td>70.1</td>
<td>65.1</td>
<td>67.5</td>
</tr>
<tr>
<td><strong>SP\textsubscript{question,BERT}</strong></td>
<td><strong>71.1</strong></td>
<td>72.6</td>
<td><strong>71.8</strong></td>
</tr>
</tbody>
</table>

**Combination using OR vs. AND.** We combine the answers to the two generated questions by an "OR" operator, but the same can be done with the "AND" operator. To check this we ran our models but report the relation as "present" iff the two questions return a correct answer. The results are reported in Table 2. The AND operator greatly underperforms when compared to the OR operator with a drop in F\textsubscript{1} of about 10%. The reason for this degradation is that the AND operator is more focused on precision, while the OR operator is more focused on recall. Over the years a major challenge of RC system was to increase recall (She et al., 2018) - It’s easier for RC system to filter unrelated samples than to generalize to new patterns.

### 8 Relation to SQuAD Training

We advocated a fully supervised training of RC models as span-prediction. How well does this...
References


Xiaoya Li, Jингrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019a. A Unified MRC Framework for Named Entity Recognition.
Tables 5, 6 and 7 on the next page show the question templates we used on the TACRED dataset for the QA reduction.
<table>
<thead>
<tr>
<th>Relation Name</th>
<th>Question</th>
</tr>
</thead>
</table>
| per:date_of_birth             | Q1: When was $e_1$ born?  
Q2 Who was born in $e_2$? |
| per:title                     | Q1: What is $e_1$’s title?  
Q2 Who has the title $e_2$ |
| org:top_members/employees     | Q1: Who are the top members of the organization $e_1$?  
Q2 What organization is $e_2$ a top member of? |
| org:country_of_headquarters   | Q1: In what country the headquarters of $e_1$ is?  
Q2 What organization have it’s headquarters in $e_2$? |
| per:parents                   | Q1: Who are the parents of $e_1$?  
Q2 Who are the children of $e_2$? |
| per:age                       | Q1: What is $e_1$’s age?  
Q2 Whose age is $e_2$? |
| per:countries_of_residence    | Q1: What country does $e_1$ resides in?  
Q2 Who resides in country $e_2$? |
| per:children                  | Q1: Who are the children of $e_1$?  
Q2 Who are the parents of $e_2$? |
| org:alternate_names           | Q1: What is the alternative name of the organization $e_1$?  
Q2 What is the alternative name of the organization $e_2$? |
| per:charges                   | Q1: What are the charges of $e_1$?  
Q2 Who was charged in $e_2$? |
| per:cities_of_residence       | Q1: What city does $e_1$ resides in?  
Q2 Who resides in city $e_2$? |
| per:origin                    | Q1: What is $e_1$ origin?  
Q2 Who originates from $e_2$? |
| org:founded_by                | Q1: Who founded $e_1$?  
Q2 What did $e_2$ found? |
| per:employee_of               | Q1: Where does $e_1$ work?  
Q2 Who is an employee of $e_2$? |
| per:siblings                  | Q1: Who is the sibling of $e_1$?  
Q2 Who is the sibling of $e_2$? |
| per:alternate_names           | Q1: What is the alternative name of $e_1$?  
Q2 What is the alternative name of $e_2$? |
| org:website                   | Q1: What is the URL of $e_1$?  
Q2 What organization have the URL $e_2$? |
| per:religion                  | Q1: What is the religion of $e_1$  
Q2 Who believe in $e_2$ |
| per:stateorprovince_of_death | Q1: Where did $e_1$ died?  
Q2 Who died in $e_2$? |
| org:parents                   | Q1: What organization is the parent organization of $e_1$?  
Q2 What organization is the child organization of $e_2$? |
| org:subsidiaries              | Q1: What organization is the child organization of $e_1$?  
Q2 What organization is the parent organization of $e_2$? |

Table 5: TACRED question templates part 1
<table>
<thead>
<tr>
<th>Relation Name</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>per:other_family</td>
<td>Q1: Who are family of $e_1$?</td>
</tr>
<tr>
<td></td>
<td>Q2 Who are family of $e_2$?</td>
</tr>
<tr>
<td>per:stateorprovinces_of_residence</td>
<td>Q1: What is the state of residence of $e_1$?</td>
</tr>
<tr>
<td></td>
<td>Q2 Who lives in the state of $e_2$?</td>
</tr>
<tr>
<td>org:members</td>
<td>Q1: Who is a member of the organization $e_1$?</td>
</tr>
<tr>
<td></td>
<td>Q2 What organization $e_2$ is member of?</td>
</tr>
<tr>
<td>per:cause_of_death</td>
<td>Q1: How did $e_1$ die?</td>
</tr>
<tr>
<td></td>
<td>Q2 How died by $e_2$?</td>
</tr>
<tr>
<td>org:member_of</td>
<td>Q1: What is the group the organization $e_1$ is member of?</td>
</tr>
<tr>
<td></td>
<td>Q2 What organization is a member of $e_2$?</td>
</tr>
<tr>
<td>org:number_of_employees/members</td>
<td>Q1: How many members does $e_1$ have?</td>
</tr>
<tr>
<td></td>
<td>Q2 What organization have $e_2$ members?</td>
</tr>
<tr>
<td>per:country_of_birth</td>
<td>Q1: In what country was $e_1$ born</td>
</tr>
<tr>
<td></td>
<td>Q2 Who was born in the country $e_2$?</td>
</tr>
<tr>
<td>org:shareholders</td>
<td>Q1: Who hold shares of $e_1$?</td>
</tr>
<tr>
<td></td>
<td>Q2 What organization does $e_2$ have shares of?</td>
</tr>
<tr>
<td>org:stateorprovince_of_headquarters</td>
<td>Q1: What is the state or province of the headquarters of $e_1$?</td>
</tr>
<tr>
<td></td>
<td>Q2 What organization’s headquarters are in the state or province $e_2$?</td>
</tr>
<tr>
<td>per:city_of_death</td>
<td>Q1: In what city did $e_1$ die?</td>
</tr>
<tr>
<td></td>
<td>Q2 Who died in the city $e_2$?</td>
</tr>
<tr>
<td>per:city_of_birth</td>
<td>Q1: In what city was $e_1$ born</td>
</tr>
<tr>
<td></td>
<td>Q2 Who was born in the city $e_2$?</td>
</tr>
<tr>
<td>per:spouse</td>
<td>Q1: Who is the spouse of $e_1$?</td>
</tr>
<tr>
<td></td>
<td>Q2 Who was born in the city $e_2$?</td>
</tr>
<tr>
<td>org:city_of_headquarters</td>
<td>Q1: Where are the headquarters of $e_1$?</td>
</tr>
<tr>
<td></td>
<td>Q2 Which organization has its headquarters in $e_2$?</td>
</tr>
<tr>
<td>per:date_of_death</td>
<td>Q1: When did $e_1$ die?</td>
</tr>
<tr>
<td></td>
<td>Q2 Who died on $e_2$</td>
</tr>
<tr>
<td>per:schools_attended</td>
<td>Q1: Which schools did $e_1$ attend?</td>
</tr>
<tr>
<td></td>
<td>Q2 Who attended $e_2$?</td>
</tr>
<tr>
<td>org:political/religious_affiliation</td>
<td>Q1: What is $e_1$ political or religious affiliation?</td>
</tr>
<tr>
<td></td>
<td>Q2 Which organization has is political or religious affiliation with $e_2$?</td>
</tr>
<tr>
<td>per:country_of_death</td>
<td>Q1: Where did $e_1$ die?</td>
</tr>
<tr>
<td></td>
<td>Q2 Who dies in $e_2$?</td>
</tr>
<tr>
<td>org:founded</td>
<td>Q1: When was $e_1$ founded?</td>
</tr>
<tr>
<td></td>
<td>Q2 Which organization was founded on $e_2$?</td>
</tr>
<tr>
<td>per:stateorprovince_of_birth</td>
<td>Q1: In what state was $e_1$ born</td>
</tr>
<tr>
<td></td>
<td>Q2 Who was born in state $e_2$?</td>
</tr>
<tr>
<td>org:dissolved</td>
<td>Q1: When was $e_1$ dissolved?</td>
</tr>
<tr>
<td></td>
<td>Q2 Which organization was dissolved in $e_2$?</td>
</tr>
</tbody>
</table>

Table 6: TACRED question templates part 2
<table>
<thead>
<tr>
<th>Relation Name</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-Container</td>
<td>Q1: Where is the ( e_1 ) stored?</td>
</tr>
<tr>
<td></td>
<td>Q2: What is stored in the ( e_2 )?</td>
</tr>
<tr>
<td>Component-Whole</td>
<td>Q1: What whole is the ( e_1 ) component of?</td>
</tr>
<tr>
<td></td>
<td>Q2: What is the component of the ( e_2 )?</td>
</tr>
<tr>
<td>Product-Producer</td>
<td>Q1: Who produces ( e_1 )?</td>
</tr>
<tr>
<td></td>
<td>Q2: What does ( e_2 ) produce?</td>
</tr>
<tr>
<td>Instrument-Agency</td>
<td>Q1: Who uses a ( e_1 )?</td>
</tr>
<tr>
<td></td>
<td>Q2: What does ( e_2 ) use?</td>
</tr>
<tr>
<td>Member-Collection</td>
<td>Q1: What collection ( e_1 ) is part of?</td>
</tr>
<tr>
<td></td>
<td>Q2: What is a fraction of ( e_2 )?</td>
</tr>
<tr>
<td>Entity-Origin</td>
<td>Q1: Where does ( e_1 ) come from?</td>
</tr>
<tr>
<td></td>
<td>Q2: What comes from ( e_2 )?</td>
</tr>
<tr>
<td>Entity-Destination</td>
<td>Q1: What is the ( e_1 )’s destination?</td>
</tr>
<tr>
<td></td>
<td>Q2: Who does ( e_2 ) serve as a destination?</td>
</tr>
<tr>
<td>Message-Topic</td>
<td>Q1: What is the topic of the ( e_1 )?</td>
</tr>
<tr>
<td></td>
<td>Q2: Who does ( e_2 ) serve as a topic?</td>
</tr>
<tr>
<td>Cause-Effect</td>
<td>Q1: What caused the ( e_1 )?</td>
</tr>
<tr>
<td></td>
<td>Q2: From what ( e_2 ) caused?</td>
</tr>
</tbody>
</table>

Table 7: SemEval question templates