Zero-Shot Information Extraction as a Unified Text-to-Triple Translation

Chenguang Wang*, **Xiao Liu**¶, **Zui Chen**¶, **Haoyun Hong**¶, **Jie Tang**¶, **Dawn Song***
*UC Berkeley, ¶Tsinghua University
{chenguangwang,dawnsong}@berkeley.edu, jietang@tsinghua.edu.cn
{liuxiao21,chenzui19,honghy17}@mails.tsinghua.edu.cn

Abstract

We cast a suite of information extraction tasks into a text-to-triple translation framework. Instead of solving each task relying on task-specific datasets and models, we formalize the task as a translation between task-specific input text and output triples. By taking the task-specific input, we enable a task-agnostic translation by leveraging the latent knowledge that a pre-trained language model has about the task. We further demonstrate that a simple pre-training task of predicting which relational information corresponds to which input text is an effective way to produce task-specific outputs. This enables the zero-shot transfer of our framework to downstream tasks. We study the zero-shot performance of this framework on open information extraction (OIE2016, NYT, WEB, PENN), relation classification (FewRel and TACRED), and factual probe (Google-RE and T-REx). The model transfers non-trivially to most tasks and is often competitive with a fully supervised method without the need for any task-specific training. For instance, we significantly outperform the F1 score of the supervised open information extraction without needing to use its training set.[1]

1 Introduction

Information extraction refers to the task of automatically extracting structured information from unstructured resources, benefiting a wide range of applications such as information retrieval and knowledge base population. Information extraction covers a great variety of tasks in natural language processing (NLP), such as open information extraction and relation classification. For example, given a sentence "Born in Glasgow, Fisher is a graduate of the London Opera Centre", open information extraction seeks to extract (Fisher; Born in; Glasgow), and "city_of_birth" is predicted as the relation between a given pair of entities "Fisher" and "Glasgow" for relation classification.

Most current approaches design task-specific pipelines for different information extraction tasks. Yet, this presents two limitations for information extraction. First, since most of the approaches employ a task-specific model, it is difficult to leverage a single pipeline to solve many tasks or adapt a model trained on one task to another without changing any task-specific modules. Second, those supervised state-of-the-arts are trained on task-specific corpora to predict from a fixed set of task-specific categories, which restricts their usability since additional labeled data is needed to specify any other classes. Such task-specific labeled data is scarce in information extraction. For example, the largest training set for open information extraction contains only 3,200 sentences (Stanovsky et al., 2018). Motivated by this, we aim to solve information extraction tasks within the same framework in a task-agnostic setting.

In this paper, we propose a unified framework for information extraction. The basic idea is to treat every information extraction problem as a "text-to-triple" problem, i.e., translating input text to output triples. We successfully apply our framework to three information extraction tasks, greatly improving zero-shot performance on many datasets and sometimes even reaching competitiveness with the current state-of-the-art fully supervised approaches. Figure 1 shows how different information extraction tasks are handled within our framework. The framework encodes task priors in the input text and decodes the output triples to finally produce task predictions. We achieve this by leveraging the same translation process on all tasks, the only difference among tasks being the input encoding. This is in contrast with previous approaches using task-specific models and datasets. The design of the common translation module for all tasks is important: by leveraging the task priors encoded in

[1]The code and datasets are available at https://github.com/cgraywang/deepex.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1225–1238
November 7–11, 2021.

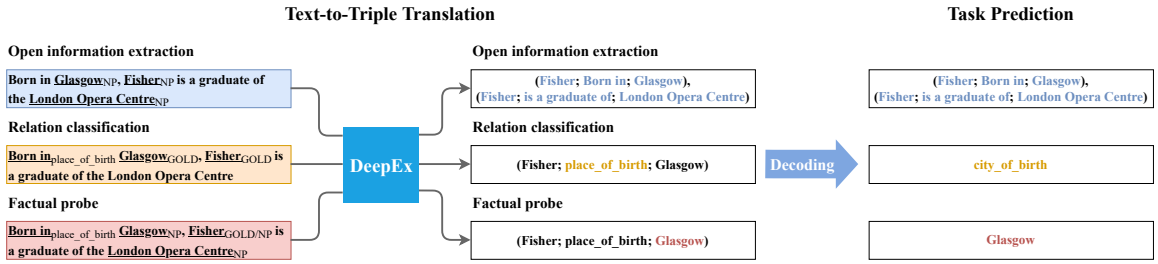


Figure 1: Our DEEPEX translates between input text and output triples, and the output is then decoded into task predictions.

the input text, we enable the zero-shot transfer of the general knowledge that a pre-trained LM has about the task. We demonstrate that a simple pre-training task of predicting which relational triple goes with which text on a task-agnostic corpus further enhances the zero-shot capabilities on all tasks. To the best of our knowledge, this is the first framework to handle a variety of information extraction tasks in a zero-shot setting.

Our contributions are summarized below.

1. We introduce DEEPEX, a unified framework that solves information extraction tasks in a zero-shot setting. We cast information extractions as text-to-triple problems by incorporating the task priors in the input text and translating the input text to triples as output.

2. We apply our framework to (i) open information extraction, (ii) relation classification, and (iii) factual probe. In all tasks, we achieve competitive zero-shot performance to the current state-of-the-art including the fully supervised methods, and we achieve new state-of-the-art performance on open information extraction (OIE2016, WEB, NYT, and PENN) and factual probe (T-REx). For instance, our zero-shot approach significantly outperforms the supervised open information extraction by averaging 37.5 points in F1.

3. We also show that our framework delivers more interpretable results while achieving comparable performance on all tasks, thanks to the transparency of the text-to-triple translation.

2 Method

We cast a suite of information extraction tasks into a text-to-triple translation framework. As shown in Figure 1, input and output are designed in a format that is appropriate for a given task. The translation process takes the input text and produces triples as output. The decoding step generates task predictions from the output. In this section, we describe the input and output format, the translation, and the decoding process. We use open information extraction (OIE) as a running example in this section. For OIE, we are given a sentence and asked to extract triples.

2.1 Input and Output Format

The input is a NP-chunked sentence, and the output is a set of triples. The NPs are encoded as task priors in the input. Below is an example.

> **Input** Born in Glasgow$_{NP}$, Fisher$_{NP}$ is a graduate of the London Opera Centre$_{NP}$.
> **Output** (Fisher; Born in; Glasgow), (Fisher; is a graduate of; London Opera Centre).

NP denotes the noun phrase.

2.2 Zero-Shot Translation

We aim to translate the above input text to output triples. Information extraction tasks lack high-quality training data, therefore training an end-to-end supervised approach (Paolini et al., 2021) is not feasible. Pre-trained language models (LM) (e.g., BERT (Devlin et al., 2019) and GPT (Brown et al., 2020)) have demonstrated their zero-shot capabilities in a wide range of NLP tasks, thanks to the general information that they know about the tasks.

We therefore propose a zero-shot translation process consisting of two steps: generating and ranking, as shown in Figure 2. The generating stage produces general information about the task via pre-trained LMs, and the ranking stage looks for specific information about the task via a ranking model pre-trained on a task-agnostic corpus.

Generating The generating stage produces a set of candidate triples that contain general information about the task from pre-trained LMs. OIE is

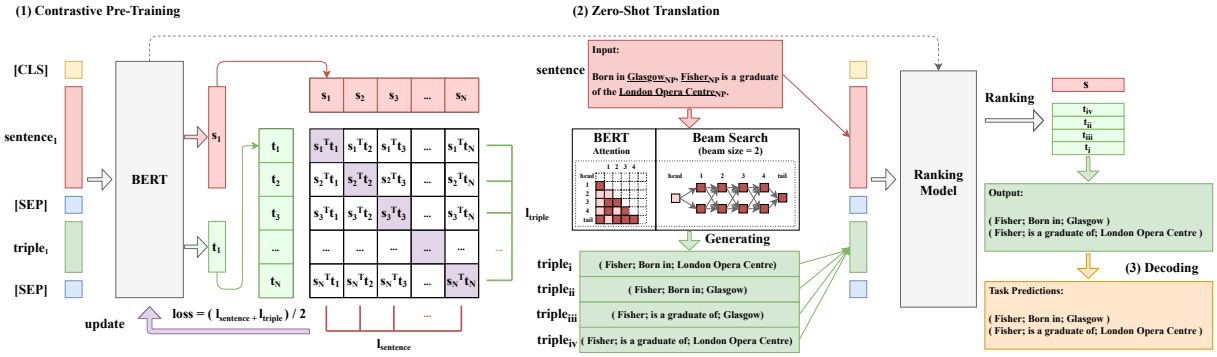


Figure 2: Summary of our approach. The framework encodes task-relevant information in the input text and decodes the output triples to produce task predictions. The zero-shot translation first generates general information that a pre-trained language model has about the input, then ranks to find the output triples of interest to the task via a ranking model pre-trained on a task-agnostic relational corpus.

formulated as extracting a set of sequences in the input that are generally relevant to an argument pair (i.e., NP pair). We particularly use the attention scores in pre-trained LMs to measure the relevance between the sequence and the argument pair.

We frame the process as a search problem. Given an argument pair (e.g., "Fisher" and "London Opera Centre"), we aim to search for the sequences (e.g., "is a graduate of") with the largest attention scores connecting the pair. To compute a score for every possible sequence is computationally expensive, especially when the sequence length is large. Therefore the exhaustive search is intractable. We use beam search, which is an approximate strategy to explore the search space efficiently. Beam search maintains the k-best candidates. This means the time cost of beam search does not depend on the sequence length but the size of the beam and the average length of the candidates. The beam search starts with a task-specific start token `[S]`. At each step, beam search simply selects top-k next tokens with the largest attention scores, and just keeps k partial candidates with the highest scores, where k is the beam size. When a candidate produces a task-specific end token `[E]`, the candidate is complete. For OIE, `[S]` and `[E]` refer to the argument pair, e.g, (`[S]` is "Fisher", and `[E]` refers to "London Opera Centre").

The above traditional beam search only allows searching sequences between `[S]` and `[E]`. To adapt beam search to produce more triples, we allow searching sequences: (i) left to both `[S]` and `[E]`, and (ii) right to both `[S]` and `[E]`. This helps to improve the recall of the candidates. For example, a candidate triple (Fisher; Born in; Glasgow) is generated by looking at "Born in" on the left in the above example. We also need to enable bidirectionality by running the search in both directions (left to right and right to left) following Wang et al. (2020). For OIE, we implement this by allowing every argument as both `[S]` and `[E]` regardless of its position in the input. For example, "Fisher" is `[S]` in (Fisher; Born in; Glasgow) given "Glasgow" appears before "Fisher" in the input.

Ranking The ranking stage finds triples that are of interest to the task via a ranking model pre-trained on a task-agnostic relational corpus. For OIE, the generating stage produces k candidate triples for every argument pair. However, the sequences in the candidates are relevant to the argument pairs, not just in the relational aspect. The ranking stage aims to find the triples that specifically express the relational information between the argument pair, which is important for OIE.

We propose a contrastive model to rank the triples as illustrated in Figure 2. Given a batch of N (sentence, triple) pairs, the model is trained to predict which of the N^2 possible (sentence, triple) pairs across a batch actually appeared. The model learns a joint embedding space by training a base encoder BERT. The input sequence of the BERT encoder is in the format: `[CLS]` sentence `[SEP]` triple `[SEP]`, which follows the standard input format of BERT. The goal is to maximize the cosine similarity of the sentence and triple embeddings of the N true pairs in the batch while minimizing the cosine similarity of the embeddings of the remaining $N^2 - N$ incorrect pairs. We optimize a cross-entropy loss over these similarity scores. The loss function for a positive pair is defined by l in Eq. 1.

$$l_{\text{sentence}} = -\log \frac{\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_i))}{\sum_{k=1}^{N} \exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_k))}$$

$$l_{\text{triple}} = -\log \frac{\exp(\text{sim}(\mathbf{u}_i, \mathbf{v}_i))}{\sum_{k=1}^{N} \exp(\text{sim}(\mathbf{u}_k, \mathbf{v}_i))}$$

$$l = \frac{l_{\text{sentence}} + l_{\text{triple}}}{2} \quad (1)$$

where $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\intercal \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$. For the i-th positive (sentence, triple) pair, $\mathbf{u}_i$ and $\mathbf{v}_i$ denote the sentence and triple embedding respectively.

We take advantage of the pre-trained $\text{BERT}_{\text{BASE}}$ as the base encoder. We further simplify the standard contrastive learning framework by removing the projection layer between the representation and the contrastive embedding space. Neither the linear (Radford et al., 2021) nor non-linear (Chen et al., 2020b) projection is used. This is because sentences and triples are unified in the same embedding space of BERT. We train the model on T-REx (Elsahar et al., 2019), which is a dataset of large-scale alignments between Wikipedia abstracts and Wikidata triples. T-REx contains a large number of sentence-triple pairs (11 million triples are paired with 6.2 million sentences). T-REx also reports an accuracy of 97.8% of the pairs.

The ranking model is task-agnostic. The ranking model takes the input in the same format for all tasks. At test time, the input text and each candidate triple from the generating stage is paired as the input to the ranking model. The candidate triples are ranked by the contrastive loss. We adopt the top-n candidate triples returned by the ranking model as the output. n varies across different tasks [2]. For the above OIE example, the output is the top-2 triples.

2.3 Decoding

Once the output triples are produced, we decode the output triples to obtain task predictions. For OIE, the output triples serve as task predictions directly. No specific decoding strategy is needed.

3 Information Extraction Tasks

3.1 Open Information Extraction

The details are provided in Sec. 2.

3.2 Relation Classification

For this task, we are given an input sentence with gold head and tail entities aiming to classify the relation type in a pre-defined category.

[2]Please refer to Appendix A for details.

Input and Output Format The input is a sentence encoded with gold head and tail entities, and linked relation phrases. The output is a triple. An example is below.

> **Input** Born in$_{\text{place_of_birth}}$ Glasgow$_{\text{GOLD}}$, Fisher$_{\text{GOLD}}$ is a graduate of the London Opera Centre.
> **Output** (Fisher; place_of_birth; Glasgow).

GOLD denotes the gold entity. The linked relation phrases annotated with Wikidata predicates, e.g., Born in$_{\text{place_of_birth}}$, are constructed as follows. We use an offline dictionary that maps the pre-defined relations to the Wikidata predicates. Such dictionaries are often provided either by Wikidata or third-parties. In all tested datasets, we can refer to either gold Wikidata or other high-quality resources for the dictionaries. We consider a sequence of tokens linked to a certain relation if the tokens are matched with the label or alias of the particular predicate in Wikidata. In the above example, "Born in" matches an alias of the Wikidata predicate "place_of_birth". In practice, some Wikidata predicates do not provide as many aliases as others. Inspired by Angeli et al. (2015), we follow the below procedure to add new aliases to resolve the imbalance issue: We first create a large candidate set of Wikipedia relations aligned to Wikidata predicates via distant supervision, then ask human annotators to filter out the wrong alignments. The remaining aligned relation phrases are added as new aliases of the Wikidata predicates.

Relation-Constrained Translation For the beam search in the generating stage of Sec. 2.2, `[S]` and `[E]` are the gold head and tail entities respectively. As the task requires the relations to be from a pre-defined category, using the beam search directly is not efficient. Allowing generating any token at each step might lead to sequences that do not match any pre-defined relations. Similar to De Cao et al. (2021), we use constrained beam search, which only decodes tokens belonging to a linked relation phrase. We take the top-n triples from the ranking model as the output.

Decoding Relation We take the Wikidata predicates of the output triples, and map the predicates back to the relations in the pre-defined category, which serve as the task predictions. In the above input/output example, "place_of_birth" is the Wikidata predicate in the output triple. It is mapped to

"city_of_birth" in the pre-defined relation category of one of the widely used relation classification datasets, TACRED. "city_of_birth" hence serves as the task prediction.

3.3 Factual Probe

Given an input sentence with gold head entity name and relation name, the task aims to fill in the tail entity.

Input and Output Format The input is encoded as a NP-chunked sentence with gold head entity candidates and linked relation phrases. The output is a triple. An example is below.

> **Input** Born in$_{\text{place_of_birth}}$ Glasgow$_{\text{NP}}$, Fisher$_{\text{GOLD/NP}}$ is a graduate of the London Opera Centre$_{\text{NP}}$.
> **Output** (Fisher; place_of_birth; Glasgow).

GOLD/NP denotes the noun phrase that matches the gold head entity. Born in$_{\text{place_of_birth}}$ represents a linked relation phrase annotated with a Wikidata predicate which is constructed in the same way as in Sec. 3.2.

Entity-Constrained Translation For the beam search, `[S]` and `[E]` are the gold head entity candidate and linked relation phrase respectively. Similar to the relation classification, we also constrain the search to generate possible tail entity sequences. We assume that NPs other than the gold head entity provide the set of candidate tail entities. To enable this, the search only decodes tokens belonging to the candidate NPs. In practice, we take the top-1 triple from the ranking model as the output.

Decoding Tail Entity We take the tail entities of the output triples as task predictions. For example, in the above output triple, "Glasgow" is decoded as the task prediction.

4 Experiments

In this section, we show that DEEPEX framework solves the different information extraction tasks considered and outperforms the task-specific state-of-the-art results on multiple datasets.

To keep the framework as simple as possible, most settings and hyperparameters are shared across all experiments. For example, we use BERT$_{\text{LARGE}}$ (Devlin et al., 2019) for the beam search of the generating stage (Sec. 2.2) for all tasks. The details of the experimental setup, datasets, and comparison methods are described in Appendix A.

4.1 Main Results

The results are shown in Table 1. We achieve state-of-the-art results on the following datasets in a zero-shot setting even outperforming fully supervised methods: (i) Open information extraction (OIE): OIE2016, WEB, NYT, PENN; and (ii) Factual probe: T-REx. The improvements are significant for OIE. In particular, the zero-shot DEEPEX outperforms RnnOIE by on average 37.5 in F1 and 44.6 in AUC, which is a supervised OIE system introduced in (Stanovsky et al., 2018). Given no specific OIE training data is used by DEEPEX, the results are encouraging, suggesting that the zero-shot transfer of the latent knowledge that a pre-trained LM has about the tasks is successful. State-of-the-art OIE performance is obtained without referring to task-specific training data, and such zero-shot ability is generalizable across multiple datasets. The PR curves for all OIE test sets are depicted in Figure 3. Similar to the findings in Table 1, overall, DEEPEX outperforms the comparison systems across all datasets. For each dataset, it provides a superior precision-recall curve. DEEPEX slightly outperforms the comparison methods on T-REx (factual probe). The main reason is that the task-specific LAMA (Petroni et al., 2020) can use the wrong memory of LMs to answer without needing the mention of the tail entity. An example expressing the triple (Nicholas Liverpool; place_of_death; Miami) is shown in Table 6 in Appendix. Thanks to the explainability and transparency of our framework, we can avoid such errors. The results demonstrate that the zero-shot DEEPEX generalizes well to different information extraction tasks.

For other datasets, we obtain comparable performance with the best comparison methods. We highlight that our approach uses a unified framework that tackles all the tasks in a zero-shot way. Our framework is task-agnostic without task-specific training or module modification, which is in contrast with task-specific models trained on specific corpora as shown in Table 1. For relation classification, all the comparison methods are fully supervised and trained on the corresponding large-scale corpora. Our top-1 zero-shot result serves as a lower bound, while top-10 results indicate an ideal situation when an improved ranking model is available. Interestingly, on FewRel, a benchmark

Open Information Extraction (F1 and AUC)

	OIE2016		WEB		NYT		PENN	
ClausIE (Del Corro and Gemulla, 2013)	58.8	37.6	44.9	40.1	29.6	22.9	34.6	28.4
Open IE4 [4]	59.6	41.7	55.7	40.5	38.3	24.0	42.6	28.1
PropS (Stanovsky et al., 2016)	55.6	33.8	58.9	48.0	37.2	22.1	39.1	27.7
RnnOIE* (Stanovsky et al., 2018)	67.0	44.5	58.1	43.5	28.3	10.4	34.5	18.3
MAMA (Wang et al., 2020)	36.6	12.8	54.3	30.3	32.9	9.4	33.0	9.0
DEEPEX (Zero-Shot) (ours)	**72.6**	**58.6**	**91.2**	**82.4**	**85.5**	**72.5**	**88.5**	**81.5**

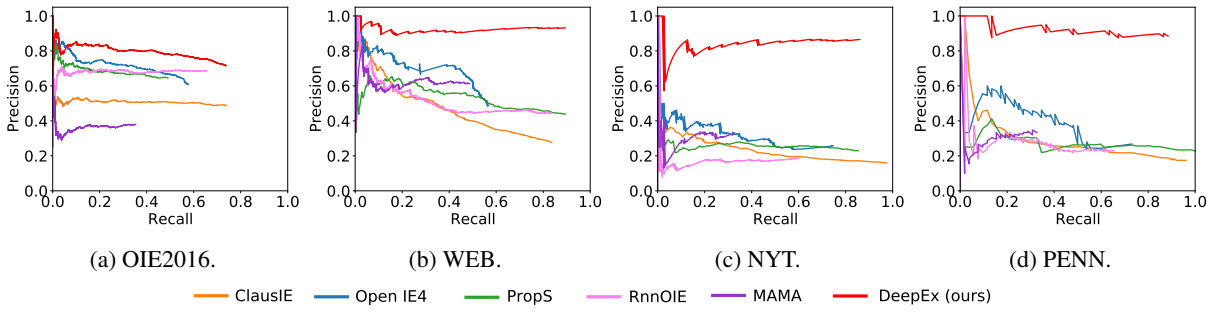
Relation Classification (F1)

	TACRED	FewRel 1.0 (dev)			
		5-way 1-shot	5-way 5-shot	10-way 1-shot	10-way 5-shot
BERT-EM* (Soares et al., 2019)	70.1	88.9	-	82.8	-
BERT_{EM}+MTB* (Soares et al., 2019)	71.5	90.1	-	83.4	-
DG-SpanBERT* (Chen et al., 2020a)	71.5	-	-	-	-
BERT-PAIR* (Gao et al., 2019)	-	85.7	89.5	76.8	81.8
TANL* (Paolini et al., 2021)	71.9	**94.0**	**96.4**	82.6	88.2
DEEPEX (Zero-Shot Top-1) (ours)	49.2	48.8	48.8	48.8	48.8
DEEPEX (Zero-Shot Top-10) (ours)	**76.4**	92.9	92.9	**92.9**	**92.9**

Factual Probe (P@1)

	Google-RE				T-REx
	birth-place	birth-date	death-place	Total	Total
LAMA-Original (Petroni et al., 2019)	16.1	1.4	14.0	10.5	32.3
LAMA-Oracle (Petroni et al., 2020)	**70.6**	**98.1**	**65.1**	**78.0**	62.6
DEEPEX (Zero-Shot) (ours)	67.8	91.0	64.1	74.3	**66.0**

Table 1: Results on all tasks. All evaluation scores are higher the better. An asterisk (*) indicates a supervised method.



(a) OIE2016. (b) WEB. (c) NYT. (d) PENN.

Figure 3: Precision-recall curves of the different open information extraction (OIE) systems on OIE datasets.

Dataset	Example (sentence and gold triple)	LAMA-Oracle	DEEPEX (ours)	Error Type
Google-RE	*Benny Marinelli (c.* **1902** *– October 22, 1927) was an American Thoroughbred horse racing jockey best known for winning the Classic Preakness Stakes in 1923 .* (Benny Marinelli; birth_date; **1902**)	1902 ✓	- ✗	Missing relation
	Myer Hoffman (21 July 1902 in **Leeds***, Yorkshire, England – 14 October 1959 in Lourenço Marques, Mozambique) was an English-born Irish cricketer.* (Myer Hoffman; place_of_birth; **Leeds**)	Leeds ✓	- ✗	Missing relation
T-REx	*..., and Bulman, as priest Frank Kane in* **BBC** *drama The Paradise Club (1989–90), and as General Tagge in the first Star Wars film (1977).* (The Paradise Club; original_network; **BBC**)	BBC ✓	- ✗	Missing relation
	Judges' lodgings, the house once occupied by former Prime Minister Edward Heath at **Salisbury**. (Edward Heath; place_of_death; **Salisbury**)	Salisbury ✓	- ✗	Wrong relation

Table 2: Error analysis of DEEPEX on factual probing datasets.

for few-shot relation classification, our top-10 zero-shot performance sometimes is the best. While TACRED is not specifically a few-shot dataset, there are many label types that rarely appear in the training set (Paolini et al., 2021). This shows the importance of zero-shot information extraction in low-resource regimes. The ranking model is based on $\text{BERT}_{\text{BASE}}$. It is interesting to check whether larger pre-trained LMs (e.g., $\text{BERT}_{\text{LARGE}}$) are more capable of ranking. We plan to investigate this in the future. On the other factual probing dataset, Google-RE, we perform slightly worse compared to LAMAs. This is mainly due to the missing mentions of relations in the sentences as shown in Table 2.

4.2 Error Analysis

To better understand the limitations of DEEPEX, we perform a detailed analysis of errors in its recall as DEEPEX lacks more in recall compared to precision across all the datasets. We use open information extraction as an example. We only show F1 and AUC in Table 1, and Figure 3 illustrates the precision-recall curves showing recall errors are the main limitation. We therefore randomly sample 50 recall errors made by DEEPEX on the WEB corpus and summarized the types of common errors as below. We find 46% of the errors are due to the spaCy noun chunker identifying the wrong arguments. 12% of the recall errors are cases where the predicate is a noun or nominalized. 10% of the examined errors are involved in long sentences. Details are described in Table 5 in Appendix.

While most of the error types are shared across the datasets, we find a type of error due to the explainability and transparency of DEEPEX, which we cannot avoid. The error is mainly due to the missing mention of relations in the sentences. This type of error mainly appears in factual probing and relation classification datasets. The reason is that the tasks do not require the existence of the actual relation span in the input. The tasks often provide the relation as an input or the relation is expressed in a vague way that can not be linked to a predicate. We take factual probe as an example in Table 2. A sentence is given to express the "place_of_death" relation can only contain mentions of "residence" relation such as "occupied by". While the gold data might consider this as a correct prediction, DEEPEX uses triples extracted from the sentences. We sacrifice performance for better explainability and transparency. We believe it is ideal to allow a trade-off between performance and explainability. We leave this as future work. Also, "birth date" can be expressed as "(c.". Again in such cases, we sacrifice performance for explainability.

Method	P	R	F1	AUC
DEEPEX (ours)	70.9	73.8	72.3	57.4
No beam search-RnnOIE	72.2	60.8	66.0	49.9
RnnOIE	70.0	66.8	68.3	47.1
In-between beam search	68.9	65.7	67.3	50.3
Beam search-$\text{BERT}_{\text{BASE}}$	71.9	66.7	69.2	53.0
No ranking model	39.6	38.1	38.8	13.8

Table 3: Ablation over different facets of DEEPEX on OIE2016 dev set.

4.3 Ablation Studies

We perform ablation experiments to understand the relative importance of different facets of DEEPEX. We demonstrate the importance of the generating stage in particular beam search and the size of the pre-trained LM, and the ranking stage in particular the ranking model on open information extraction. We evaluate the below settings on the OIE2016 dev set. The first three settings examine the relative importance of the beam search: (i) No beam search-RnnOIE: Instead of using the beam search, we use the best supervised OIE system RnnOIE in the generating stage. All the other components of DEEPEX including the ranking model, are kept the same. We add the results of RnnOIE as well for comparison. (ii) In-between beam search: We only allow searching sequences between the argument pair while keeping the other components of DEEPEX the same as the default. (iii) Beam search-$\text{BERT}_{\text{BASE}}$: Instead of $\text{BERT}_{\text{LARGE}}$, we use $\text{BERT}_{\text{BASE}}$ for beam search. (iv) No ranking model: We do not leverage the ranking model, and directly rank the candidates using their original scores based on the attention from the generating stage.

We first examine the effect brought by the beam search. As shown in Table 3, we find removing beam search of the generating stage greatly hurts the performance. DEEPEX outperforms the best supervised OIE system by 6.3 in F1 and 7.5 in AUC. The result confirms our intuition that pre-trained LMs enable the zero-shot transfer of the latent knowledge that they have about the task. The original RnnOIE performs similarly; this is due to the training set of RnnOIE which provides good

coverage of the triples on the dev set. We secondly study the importance of the triple-oriented beam search. We find limiting the search significantly hurts the performance. It is often that the triples are expressed in inverted sentences, such as (Fisher; Born in; Glasgow) from "Born in Glasgow, Fisher is a graduate of the London Opera Centre". In fact, a considerable amount of gold triples containing valid relation sequences appear outside the argument pair. For example, 16.9% of the relation sequences are not between the argument pairs on the OIE2016 test set. More results are shown in Appendix B. We then test the impact of the size of the pre-trained LM. We find that $\text{BERT}_{\text{BASE}}$ performs worse than $\text{BERT}_{\text{LARGE}}$. This indicates that larger pre-trained LMs (e.g., $\text{BERT}_{\text{LARGE}}$) provide more general knowledge about the task that improves the results. Next, we study the impact of the ranking model. We find that removing the ranking model significantly hurts the performance. The results suggest that the ranking model can distinguish the relational triples from the rest among the candidates.

5 Related Work

Relation classification aims to identify the correct relation type from a pre-defined set of relations between two given entities. Language models (LM) (Han et al., 2021) pre-trained with self-supervised (Liu et al., 2021a) objectives, e.g., BERT (Devlin et al., 2019), GPT (Radford et al., 2018, 2019; Brown et al., 2020), RoBERTa (Liu et al., 2019), transfer well to relation classification datasets in fine-tuning (Joshi et al., 2020; Gao et al., 2019) or few-shot regime (Soares et al., 2019) with architecture modifications. Sequence-to-sequence models, such as T5 (Raffel et al., 2020), BART (Lewis et al., 2020) and GLM (Du et al., 2021), are adapted to the task based on data augmentation and fine-tuning (Paolini et al., 2021). Besides relation classification, Paolini et al. (2021) generalize T5 to some more structured prediction tasks as well, e.g., semantic role labeling and event extraction. However, DEEPEX enables zero-shot relation classification that does not require any task-specific training.

Many open information extraction (OIE) systems, e.g., Stanford OpenIE (Angeli et al., 2015), OLLIE (Schmitz et al., 2012), Reverb (Fader et al., 2011), and their descendant Open IE4 leverage carefully-designed linguistic patterns (e.g., based on dependencies and POS tags) to extract triples from textual corpora without using additional training sets. Recently, supervised OIE systems (Stanovsky et al., 2018; Ro et al., 2020; Kolluru et al., 2020) formulate the OIE as a sequence generation problem using neural networks trained on additional training sets. Similar to our work, Wang et al. (2020) use the parameters of LMs to extract triples, with the main difference that DEEPEX not only improves the recall of the beam search, but also uses a pre-trained ranking model to enhance the zero-shot capability.

LMs are used in factual probing tasks, by using the outputs alone (Petroni et al., 2019) to answer the relation-specific queries in cloze statements. Petroni et al. (2020) additionally feed sentences expressing the facts to the LMs and shows improved results. Other than template-based queries, learning trigger-based (Shin et al., 2020) and continuous prompts (Liu et al., 2021b; Li and Liang, 2021) are helpful in recalling the facts. The main difference is that DEEPEX explores the internal parameters of the LMs rather than the outputs, and the results are more interpretable.

Overall, in contrast to the existing approaches, DEEPEX unifies the open information extraction, relation classification, and factual probe under the same framework in zero-shot settings.

6 Conclusion

We have demonstrated that our unified approach can handle multiple information extraction tasks within a simple framework and shows improvements in zero-shot settings. Unlike previous approaches designing complicated task-specific pipelines, DEEPEX enables conducting all considered information extraction tasks with only input and output design. Therefore, DEEPEX is flexible and can be adapted to a variety of tasks. Different from previous approaches that target pre-defined categories (e.g., fixed relation types for relation classification), DEEPEX generalizes better to unseen classes as the generating stage leverages the transfer of latent knowledge that a pre-trained language model has about the tasks. Besides, the ranking stage pre-trains on a large-scale task-agnostic dataset. DEEPEX exhibits strong zero-shot capabilities in low-resource tasks without the need of any task-specific training set. DEEPEX also exploits the in-depth information of the language models, i.e., parameters, rather than the outputs alone, which en-

hances the explainability through enhanced model transparency. Based on our findings, we believe that the unified approach advances the research in understanding natural language semantics (e.g., structure prediction tasks) using deep learning models. We hope our results will foster further research in this direction.

Acknowledgement

We would like to thank the anonymous reviewers for their suggestions and comments. This material is in part based upon work supported by Berkeley DeepDrive and Berkeley Artificial Intelligence Research.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*, pages 344–354.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. Carb: A crowdsourced benchmark for open ie. In *EMNLP*, pages 6263–6268.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901.

Jun Chen, Robert Hoehndorf, Mohamed Elhoseiny, and Xiangliang Zhang. 2020a. Efficient long-distance relation extraction with dg-spanbert. *arXiv preprint arXiv:2004.03636*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *ICLR*.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *WWW*, pages 355–366.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. All nlp tasks are generation tasks: A general pretraining framework. *arXiv preprint arXiv:2103.10360*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Elena Simperl, and Frederique Laforest. 2019. T-rex: A large scale alignment of natural language with knowledge base triples.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *EMNLP*, pages 1535–1545.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel 2.0: Towards more challenging few-shot relation classification. In *EMNLP*, pages 6250–6255.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. 2021. Pre-trained models: Past, present and future. *AI Open*.

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 *EMNLP*, pages 4803–4809.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *EMNLP*, pages 643–653.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Soumen Chakrabarti, et al. 2020. Openie6: Iterative grid labeling and coordination analysis for open information extraction. *arXiv preprint arXiv:2010.03147*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021a. Self-supervised learning: Generative or contrastive. *TKDE*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *EMNLP*, pages 447–457.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *ICLR*.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *AKBC*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *EMNLP*, pages 2463–2473.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, (8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21:1–67.

Youngbin Ro, Yukyung Lee, and Pilsung Kang. 2020. Multi^2OIE: Multilingual open information extraction based on multi-head attention with BERT. In *Findings of EMNLP*, pages 1107–1117.

Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *EMNLP*, pages 523–534.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Eliciting knowledge from language models using automatically generated prompts. In *EMNLP*, pages 4222–4235.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *ACL*, pages 2895–2905.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *EMNLP*, pages 2300–2305.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with props. *arXiv preprint arXiv:1603.01648*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *NAACL*, pages 885–895.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*, pages 38–45.

Ying Xu, Mi-Young Kim, Kevin M Quinn, Randy Goebel, and Denilson Barbosa. 2013. Open information extraction with tree kernels. In *NAACL*, pages 868–877.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*, pages 35–45.

A Experimental Setup

In all experiments, for the generating stage in Sec. 2.2, we use a pre-trained $\text{BERT}_{\text{LARGE}}$ model (Devlin et al., 2019) for the beam search. In particular, we use the mean operation over the multi-head attention weights from the last layer of $\text{BERT}_{\text{LARGE}}$ according to the parameter study in (Wang et al., 2020). For the ranking model in Sec. 2.2, we use the pre-trained $\text{BERT}_{\text{BASE}}$ model. We use the implementations of the pre-trained language models (LM) in the Transformers package (Wolf et al., 2020).

To keep our framework simple, we use the same hyperparameters across the majority of our experiments. For generating, we use: 8 GeForce RTX 2080 Ti GPUs with a batch size of 16 per GPU; maximum sequence length as 256 tokens. For datasets that provide examples beyond sentence level, we adopt spaCy sentencizer [3] to segment the texts into sentences, and each sample in the batch is a sentence. For ranking, we experiment with: 1 NVIDIA Tesla V100 GPU with a batch size of 8 per GPU; the AdamW optimizer (Kingma and Ba, 2015); linear learning rate decay starting from 10^{-6}; maximum sequence length as 512 tokens. The top-1 triple from ranking model is returned except for OIE2016 (open information extraction) since it provides a dev set. We additionally report results of top-10 triples for relation classification (FewRel and TACRED).

In the rest of this section, we describe datasets, comparison methods, additional implementation details for each information extraction task, as well as more experimental insights. Results of all experiments are in Table 1. We use the default evaluation metrics for each task as below. We show more input and output formats on all datasets in Table 7.

A.1 Open Information Extraction

Datasets We evaluate the performance of the open information extraction (OIE) systems on OIE benchmark datasets consisting of **OIE2016** (Stanovsky and Dagan, 2016), a dataset from Newswire and Wikipedia automatically converted from QA-SRL (He et al., 2015); three news datasets **NYT**, **WEB** (Mesquita et al., 2013), **PENN** (Xu et al., 2013). The statistics of the benchmark is shown in Table 4.

Dataset	Domain	#Sents	#Triples		
			Train	Dev	Test
OIE2016	News,Wiki	3,200	5,078	1,673	1,730
WEB	News,Web	500	-	-	461
NYT	News,Wiki	222	-	-	150
PENN	Mixed	100	-	-	52

Table 4: Statistics of OIE benchmark datasets.

Evaluation Methodology We follow typical OIE metrics to evaluate the systems. First, we report precision, recall, and F1 score using a confidence threshold optimized on the development set. Second, we compute a precision-recall curve by evaluating the performance of the systems at different confidence thresholds. Third, we also measure the area under the PR curve (AUC) to evaluate the overall performance of a system. To compute the above metrics, we need to match the system extractions with the gold extractions. Regarding the matching functions, we adopt the function of OIE2016 to evaluate OIE2016, NYT, WEB, and PENN.

Comparison Methods We compare our method DEEPEX to the following prominent OIE systems recently evaluated in (Stanovsky et al., 2018): ClausIE (Del Corro and Gemulla, 2013), Open IE4 [4], PropS (Stanovsky et al., 2016), RnnOIE (Stanovsky et al., 2018). We also compare to MAMA with $\text{BERT}_{\text{LARGE}}$ recently introduced in (Wang et al., 2020) that also leverages pre-trained LMs to extract open triples.

Implementation Details For input text encoding, we use spaCy noun chunks [5] to identify the NPs. We use: beam size equals to 6; the top-1 triple from the ranking model for evaluation except on OIE2016. Instead, we use top-3 triples on OIE2016 based on the parameter study on its dev set in Figure 4. Experimenting on more OIE benchmark datasets such as CaRB (Bhardwaj et al., 2019) is an interesting future direction to explore.

A.2 Relation Classification

Datasets and Metrics We evaluate on FewRel (Han et al., 2018) and TACRED (Zhang et al., 2017).

- **FewRel** contains 100 relations with 7 instances for each relation. The standard evaluation for this benchmark uses few-shot N-way

[3] https://spacy.io/api/sentencizer

[4] https://github.com/dair-iitd/OpenIE-standalone

[5] https://spacy.io/usage/linguistic-features/#noun-chunks

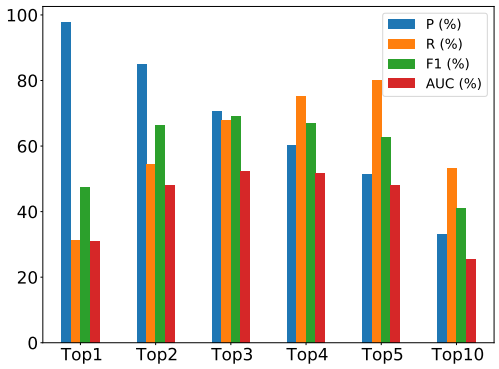


Figure 4: Effect of the ranking model on the OIE2016 dev set.

K-shot settings. The entire dataset is split into train (64 relations), validation (16 relations) and test set (20 relations). We report the same results on the dev set for all the settings because of our zero-shot setting.

- **TACRED** is a large-scale relation classification benchmark that consists of 106,344 examples and 41 relation types including 68,164 for training, 22,671 for validation, and 15,509 for testing. We do not use train and validation sets, and report the result on the test set.

We use F1 to evaluate the results.

Comparison Methods We compare our method with the following supervised methods. (i) BERT-PAIR (Gao et al., 2019) is a sequence classification model based on BERT, optimizing the score of two instances expressing the same relation. (ii) BERT_{EM} + Matching the Blanks (MTB) (Soares et al., 2019), which uses entity markers (BERT_{EM}) and additional pre-training of relations on a large-scale corpus (i.e., MTB). (iii) TANL (Paolini et al., 2021) is a sequence to sequence model based on T5 (Raffel et al., 2020) aiming to generate structured objects from an encoded natural language format.

Implementation Details For input text encoding, we attach the given gold head and tail entities to the input. As described in Sec. 3.2, linked relation phrases are also attached to the input. For TACRED, we use the relation map provided in (Angeli et al., 2015) to link relation phrases to TACRED relations, and manually build a map between TACRED relations and Wikidata predicates. For FewRel, we directly link relation phrases to gold Wikidata predicates as FewRel uses Wikidata predicates as the target category. At test time, DEEPEX makes prediction as below: given a returned triple, we first map the relation phrase of the triple back to a Wikidata predicate, then use the Wikidata predicate to find a relation in the predefined category. We report the results using two setups: top-1 and top-10 triples from the ranking model. In practice, some of the Wikidata predicates do not have aliases. We therefore follow (Angeli et al., 2015; Wang et al., 2020) to manually add aliases for such predicates based on the alignment between Wikipedia and Wikidata. We set the beam size as 6.

A.3 Factual Probe

Datasets and Metrics We consider the Google-RE consisting of 3 relations and 5,527 facts, and T-REx with 41 relations and 34,039 facts of the LAMA benchmark (Petroni et al., 2019). We evaluate the results using mean precision at one (P@1), where higher values are better.

Comparison Methods We compare to pre-trained LM based methods that leverage the output probabilities of the LM to make predictions given the sentence known to express the fact. Two methods are considered: (i) LAMA (Petroni et al., 2019) leverages the input sentence without the tail entity to query the LMs, and (ii) LAMA-Oracle (Petroni et al., 2020) enriches the query with (at most) five gold sentences as additional context.

Implementation Details For input encoding, we use spaCy noun chunks [5] to identify the NPs. We additionally use Stanford NER [6] to label the missing entities from noun chunks such as dates. A NP is considered as a gold head entity if it overlaps with the mention of the gold head entity. Therefore there might be multiple NPs representing the gold head entity in the input. Similar to relation classification, linked relation phrases are attached to the input. We use the gold mapping to convert Freebase predicates to Wikidata ones for Google-RE, and use the gold Wikidata predicates for T-REx as T-REx builds based on Wikidata. At test time, a prediction is made: given a returned triple, we conduct an exact match between the tail entity of the triple and the gold tail entity. We use the top-1

[6] https://stanfordnlp.github.io/CoreNLP/ner.html

Error type	Percentage	Example (sentence and gold triple)
Wrong NP	46%	*San Jose , Calif. - based Adobe announced in April the plans to* **acquire** *Macromedia , which makes the Flash animation software used to display graphics on Web sites .* (Adobe; **acquire**; Macromedia)
Unformatted Sentence	14%	*Adobe* **acquired** *Macromedia! - sephiroth.it - flash & php* (Adobe; **acquired**; Macromedia!)
Nominalization	12%	*Google confirms YouTube* **acquisition** *- BBC News* (Google; **acquisition**; YouTube)
Long Sentence	10%	*Amid all the hubbub over Google 's* **swallowing** *of YouTube , we ve heard both considered commentary and over - the - top pontification about whether it's a good deal or a bad deal , for the newly - engorged company and for all those users in TV land .* (Google; **swallowing**; YouTube)
Noun	4%	*Dean Kamen (left) ,* **inventor** *of the Segway Human Transporter Human Transporter wearing the Plantronics Voyager 510 Bluetooth Headset* (Dean Kamen; **inventor**; the Segway Human Transporter)

Table 5: Analysis of frequently-occurring recall errors of DEEPEX on a random sample of 50 sentences on the OIE task. For each type we list the percentage of sentences in which it occurs, and an example taken from the WEB corpus.

Dataset	Example (sentence and gold triple)	LAMA-Oracle	DEEPEX (ours)	Error Type
T-REx	*Naomi Shihab Nye (born March 12, 1952) is a* **poet**, *songwriter, and novelist.* (Naomi Shihab Nye; occupation; **poet**)	teacher ✗	poet ✓	Wrong memory
	Nicholas Liverpool died on 1 June 2015 in **Miami**, *where he was receiving medical treatment.* (Nicholas Liverpool; place_of_death; **Miami**)	London ✗	Miami ✓	Wrong memory
	Jean-Michel Pilc (born 1960 in Paris, France) is a self-taught **jazz** *pianist currently residing in New York.* (Jean-Michel Pilc; genre; **jazz**)	classical ✗	jazz ✓	Wrong memory
	Nick Lucas's version, released on **Brunswick**, *was a No.* (Nick Lucas; record_label; **Brunswick**)	EMI ✗	Brunswick ✓	Wrong memory

Table 6: Out-of-context predictions of LAMA-Oracle (Petroni et al., 2020) for the factual probing task.

Dataset	Input	Output
OIE2016	$\underline{\text{He}}_{\text{NP}}$ hasn't been able to replace $\underline{\text{the M'Bow cabal}}_{\text{NP}}$.	(He; hasn't been able to replace; the M'Bow cabal)
WEB	$\underline{\text{Crick}}_{\text{NP}}$ received $\underline{\text{a Nobel Prize}}_{\text{NP}}$ for discovering $\underline{\text{the structure}}_{\text{NP}}$ of $\underline{\text{DNA}}_{\text{NP}}$.	(Crick; received a Nobel Prize for; the structure), (DNA; for discovering the structure; a Nobel Prize)
NYT	PRODUCER – $\underline{\text{Squier Knapp Dunn Communications}}_{\text{NP}}$ in consultation with $\underline{\text{David Garth}}_{\text{NP}}$	(David Garth; in consultation with; Squier Knapp Dunn Communications)
PENN	$\underline{\text{A spokeswoman}}_{\text{NP}}$ said $\underline{\text{Sulka}}_{\text{NP}}$ operates $\underline{\text{a total}}_{\text{NP}}$ of seven stores in $\underline{\text{the U.S.}}_{\text{NP}}$ and overseas .	(Sulka; operates a total of; the U.S), (A spokeswoman; said; the U.S), (a total; of seven stores; the U.S)
TACRED	$\underline{\text{Denise Maloney Pictou}}_{\text{GOLD}}$, one of $\underline{\text{Aquash}}_{\text{GOLD}}$'s $\underline{\text{daughters}}_{\text{child}}$, says she hopes Graham 's trial will help bring justice to her family .	(Denise Maloney Pictou; child; Aquash)
FewRel 1.0 (dev)	$\underline{\text{Theodore II Palaiologos}}_{\text{GOLD}}$ was a son of the Eastern Roman Emperor Manuel II Palaiologos and his $\underline{\text{wife}}_{\text{spouse}}$ $\underline{\text{Helena Draga}}_{\text{GOLD}}$.	(Theodore II Palaiologos; spouse; Helena Draga)
Google-RE	$\underline{\text{Peter F Martin}}_{\text{GOLD/NP}}$ ($\underline{\text{born}}_{\text{birth\_date}}$ $\underline{\text{1941}}_{\text{NP}}$) is an $\underline{\text{American politician}}_{\text{NP}}$ who is a $\underline{\text{Democratic member}}_{\text{NP}}$ of the Rhode Island House of $\underline{\text{Representatives}}_{\text{GOLD/NP}}$.	(Peter F Martin; date_of_birth; 1941)
T-REx	$\underline{\text{Antonio Agliardi}}_{\text{GOLD/NP}}$ in 4 September 1832 – 19 March 1915 $\underline{\text{was an Italian}}_{\text{position\_held}}$ $\underline{\text{Roman Catholic Cardinal}}_{\text{NP}}$, $\underline{\text{archbishop}}_{\text{NP}}$, and $\underline{\text{papal diplomat}}_{\text{NP}}$.	(Antonio Agliardi; position_held; diplomat)

Table 7: Input/output examples for all datasets.

Dataset	#Triples Left Relation	Right Relation	Middle Relation	Total
OIE2016	128	165	1,437	1,730
WEB	17	29	415	461
NYT	2	29	119	150
PENN	2	4	46	52

Table 8: Statistics of relation positions of the gold triples in all OIE datasets.

System	#Triples	Words/Arg
Gold	1,730	5.38
ClausIE	2,768	5.78
Open IE4	1,793	4.55
PropS	1,551	5.80
RnnOIE	1,993	4.68
MAMA	1751	2.56
DEEPEX (ours)	1755	2.30

Table 9: Output statistics of the different OIE systems on OIE2016.

returned triple for P@1 evaluation. The beam size equals 20.

B Additional Analysis of Results

Error Analysis of OIE We show detailed error analysis of DEEPEX on the OIE task in Table 5.

Relation Position Distribution We show the statistics of relation positions in the gold triples in all OIE datasets. Overall, there is a considerable amount of triples containing relation not between the entity pair across all OIE datasets as shown in Table 8, demonstrating the importance of triple-oriented beam search. In particular, 14.8% of the triples contains relations outside the entity pairs on OIE datasets (OIE2016, WEB, NYT, PENN).

Output Statistics Table 9 compares the statistics of the outputs of different OIE systems and gold data on OIE2016. We find that DEEPEX produces 25 more triples than the gold data, and the arguments tend to be shorter.

Out-of-Context Cases for LAMA-Oracle Table 6 shows major error cases in LAMA-Oracle when compared with DEEPEX, where LAMA-Oracle often generates out-of-context answers due to the wrong memory of LMs. This shows that explainable extraction like DEEPEX is necessary for information extraction using LMs.

C Analysis of Ranking Model

Implementation Details We remove triples with empty predicates from the original T-REx (Elsahar et al., 2019), and further remove sentence-triple pairs if the triples are in the LAMA version of T-REx (Petroni et al., 2019). This results in approximately 4 million positive sentence-triple pairs. We set the number of fine-tuning epochs to 1. Each batch takes approximately 7 minutes, resulting in a total 12 hours for fine-tuning the ranking model.

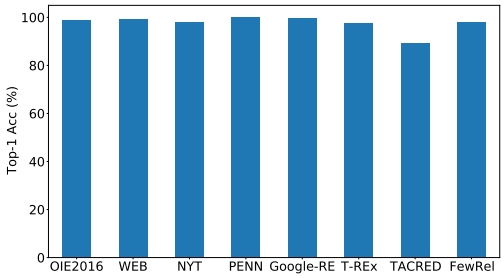


Figure 5: Top-1 accuracy of the ranking on all datasets.

Ranking Results We evaluate the performance of the ranking model. To do so, we construct negative sentence-triple samples for all the tested datasets as follows: for each sentence in a dataset, we randomly sample a triple from other sentences to construct a negative sample. We then directly evaluate the ranking performance of the model on all the datasets, and report the top-1 accuracy in Figure 5. We find that the ranking model works extremely well, obtaining nearly perfect top-1 accuracy on all the datasets.