GRIT: Generative Role-filler Transformers
for Document-level Event Entity Extraction

Xinya Du   **Alexander M. Rush**   **Claire Cardie**
Department of Computer Science
Cornell University
{xdu, cardie}@cs.cornell.edu
arush@cornell.edu

Abstract

We revisit the classic problem of document-level role-filler entity extraction (REE) for template filling. We argue that sentence-level approaches are ill-suited to the task and introduce a generative transformer-based encoder-decoder framework (GRIT) that is designed to model context at the document level: it can make extraction decisions across sentence boundaries; is *implicitly* aware of noun phrase coreference structure, and has the capacity to respect cross-role dependencies in the template structure. We evaluate our approach on the MUC-4 dataset, and show that our model performs substantially better than prior work. We also show that our modeling choices contribute to model performance, e.g., by implicitly capturing linguistic knowledge such as recognizing coreferent entity mentions.

1 Introduction

Document-level template filling (Sundheim, 1991, 1993; Grishman and Sundheim, 1996) is a classic problem in information extraction (IE) and NLP (Jurafsky and Martin, 2014). It is of great importance for automating many real-world tasks, such as event extraction from newswire (Sundheim, 1991). The complete task is generally tackled in two steps. The first step detects events in the article and assigns templates to each of them (template recognition); the second step performs role-filler entity extraction (REE) for filling in the templates. In this work we focus on the **role-filler entity extraction (REE)** sub-task of template filling (Figure 1).[1] The input text describes a bombing event; the goal is to identify the entities that fill any of the roles associated with the event (e.g., the perpetrator, their organization, the weapon) by extracting

---

[1] In this work, we assume there is one generic template for the entire document (Huang and Riloff, 2011, 2012).

Input document:

...
A bomb exploded in a Pilmai alley destroying some [**water pipes**].

According to unofficial reports, the bomb contained [**125 to 150 grams of TnT**] and was placed in the back of the [**Pilmai** [**telephone company building**]].

The explosion occurred at 2350 on 16 January, causing panic but no casualties.

The explosion caused damages to the [telephone company offices]. It also destroyed a [**public telephone booth**] and [water pipes].

Witnesses reported that the bomb was planted by [[**two men**] wearing sports clothes], who escaped into the night.
...
They were later identified as [[**Shining Path**] members].

Gold extractions:

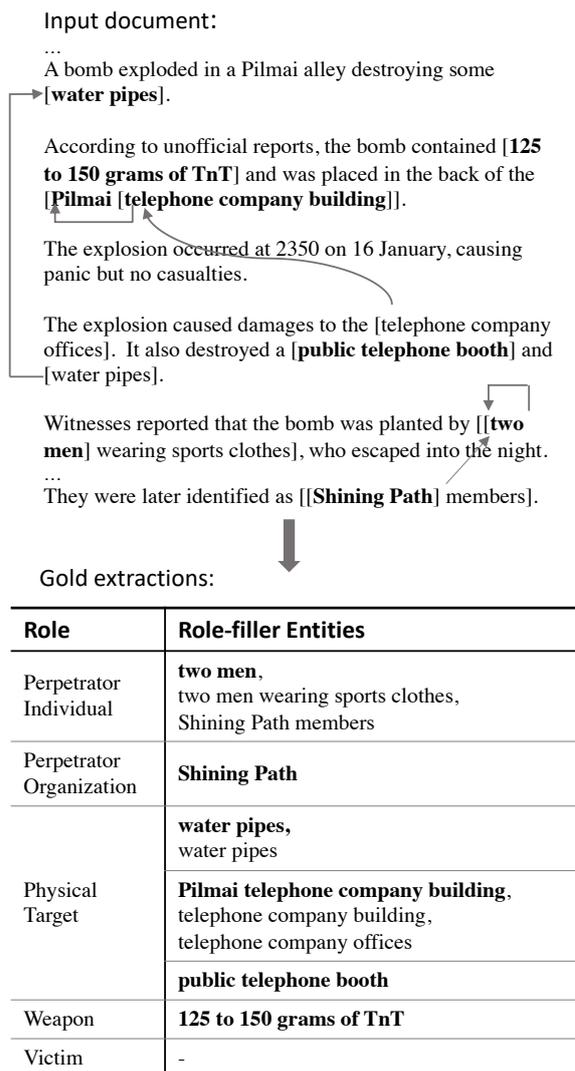| Role	Role-filler Entities
Perpetrator Individual	**two men**, two men wearing sports clothes, Shining Path members
Perpetrator Organization	**Shining Path**
Physical Target	**water pipes,** water pipes **Pilmai telephone company building**, telephone company building, telephone company offices **public telephone booth**
Weapon	**125 to 150 grams of TnT**
Victim	-

Figure 1: Role-filler entity extraction (REE). The first mention of each role-filler entity is bold in the table and document. The arrows denote coreferent mentions.

a descriptive "mention" of it – a string from the document.

In contrast to sentence-level event extraction (see, e.g., the ACE evaluation (Linguistic Data Consortium, 2005)), document-level REE introduces

several complications. First, **role-filler entities must be extracted even if they never appear in the same sentence as an event trigger**. In Figure 1, for example, the WEAPON and the first mention of the telephone company building (TARGET) appear in a sentence that does not explicitly mention the explosion of the bomb. In addition, **REE is ultimately an entity-based task** — exactly one descriptive mention for each role-filler should be extracted even when the entity is referenced multiple times in connection with the event. The final output for the bombing example should, therefore, include just one of the "water pipes" references, and one of the three alternative descriptions of the PERPIND and the second TARGET, the telephone company building. As a result of these complications, end-to-end sentence-level event extraction models (Chen et al., 2015; Lample et al., 2016), which dominate the literature, are ill-suited for the REE task, which calls for models that encode information and track entities across a longer context.

Fortunately, neural models for event extraction that have the ability to model longer contexts have been developed. Du and Cardie (2020), for example, extend standard contextualized representations (Devlin et al., 2019) to produce a document-level sequence tagging model for event argument extraction. Both approaches show improvements in performance over sentence-level models on event extraction. Regrettably, these approaches (as well as most sentence-level methods) handle each candidate role-filler prediction in isolation. Consequently, they cannot easily **model the coreference structure required to limit spurious role-filler mention extractions**. Nor can they easily **exploit semantic dependencies between closely related roles** like the PERPIND and the PERPORG, which can share a portion of the same entity span. "Shining Path members", for instance, describes the PERPIND in Figure 1, and its sub-phrase, "Shining Path", describes the associated PERPORG.

Contributions In this work we revisit the classic but recently under-studied problem of document-level role-filler entity extraction problem and introduce a novel end-to-end generative transformer model — the "**G**enerative **R**ole-filler **T**ransformer" (**GRIT**) (Figure 2).

- Designed to model context at the document level, GRIT (1) has the ability to make extraction decisions across sentence boundaries; (2) is implicitly aware of noun phrase coreference structure;

and (3) has the capacity to respect cross-role dependencies. More specifically, GRIT is built upon the pre-trained transformer model (BERT): we add a pointer selection module in the decoder to permit access to the entire input document, and a generative head to model document-level extraction decisions. In spite of the added extraction capability, GRIT requires no additional parameters beyond those in the pre-trained BERT.

- To measure the model's ability to both extract entities for each role, and implicitly recognize coreferent relations between entity mentions, we design a metric (CEAF-REE) based on a maximum bipartite matching algorithm, drawing insights from the CEAF (Luo, 2005) coreference resolution measure.

- We evaluate GRIT on the MUC-4 (1992) REE task (Section 3). Empirically, our model outperforms substantially strong baseline models. We also demonstrate that GRIT is better than existing document-level event extraction approaches at capturing linguistic properties critical for the task, including coreference between entity mentions and cross-role extraction dependencies.[2]

2 Related Work

Sentence-level Event Extraction Most work in event extraction has focused on the ACE sentence-level event task (Walker et al., 2006), which requires the detection of an event trigger and extraction of its arguments from within a single sentence. Previous state-of-the-art methods include Li et al. (2013) and Li et al. (2015), which explored a variety of hand-designed features. More recently, neural network based models such as recurrent neural networks (Nguyen et al., 2016; Feng et al., 2018), convolutional neural networks (Nguyen and Grishman, 2015; Chen et al., 2015) and attention mechanisms (Liu et al., 2017, 2018) have also been shown to help improve performance. Beyond the task-specific features learned by the deep neural models, Zhang et al. (2019) and Wadden et al. (2019) also utilize pre-trained contextualized representations.

Only a few models have gone beyond individual sentences to make decisions. Ji and Grishman (2008) and Liao and Grishman (2010) utilize event type co-occurrence patterns to propagate

---

event classification decisions. Yang and Mitchell (2016) propose to learn within-event (sentence) structures for jointly extracting events and entities within a document context. Similarly, from a methodological perspective, our GRIT model also learns structured information, but it learns the dependencies between role-filler entity mentions and between different roles. Duan et al. (2017) and Zhao et al. (2018) leverage document embeddings as additional features to aid event detection. Although the approaches above make decisions with cross-sentence information, their extractions are still done the sentence level.

Document-level IE   Document-level event role-filler *mention* extraction has been explored in recent work, using hand-designed features for both local and additional context (Patwardhan and Riloff, 2009; Huang and Riloff, 2011, 2012), and with end-to-end sequence tagging based models with contextualized pre-trained representations (Du and Cardie, 2020). These efforts are the most related to our work. The key difference is that our work focuses on a more challenging, and more realistic, setting: extracting **role-filler entities** rather than lists of role-filler mentions that are not grouped according to their associated entity. Also on a related note, Chambers and Jurafsky (2011), Chambers (2013), and Liu et al. (2019) work on *unsupervised* event schema induction and open-domain event extraction from documents. The main idea is to group entities corresponding to the same role into an event template.

Recently, there has also been increasing interest in cross-sentence/document-level relation extraction (RE). In the scientific domain, Peng et al. (2017); Wang and Poon (2018); Jia et al. (2019) study N-ary cross-sentence RE using distant supervision annotations. Luan et al. (2018) introduce SciERC dataset and their model rely on multi-task learning to share representations between entity span extraction and relations. Yao et al. (2019) construct an RE dataset of cross-sentence relations on Wikipedia paragraphs. Ebner et al. (2020) introduce RAMS dataset for multi-sentence argument *mention* linking, while we focus on entity-level extraction in our work. Different from work on joint modeling (Miwa and Bansal, 2016) and multi-task learning (Luan et al., 2019) setting for extracting entities and relations, through the generative modeling setup, our GRIT model *implicitly* captures (non-)coreference relations between noun phrases,

without relying on the cross-sentence coreference and relation annotations during training.

Neural Generative Models with a Shared Module for Encoder and Decoder   Our GRIT model uses one shared transformer module for both the encoder and decoder, which is simple and effective. For the machine translation task, He et al. (2018) propose a model which shares the parameters of each layer between the encoder and decoder to regularize and coordinate the learning. Dong et al. (2019) presents a new unified pre-trained language model that can be fine-tuned for both NLU and NLG tasks. Similar to our work, they also introduce different masking strategies for different kinds of tasks (see Section 5).

3   The Role-filler Entity Extraction Task and Evaluation Metric

We base the REE task on the original MUC[3] formulation (Sundheim, 1991), but simplify it as done in prior research (Huang and Riloff, 2012; Du and Cardie, 2020). In particular, we assume that one generic template should be produced for each document: for documents that recount more than one event, the extracted role-filler entities for each are merged into a single event template. Second, we focus on entity-based roles with string-based fillers[4].

- Each *event* consists of the set of roles that describe it (shown in Figure 1). The MUC-4 dataset that we use consists of ∼1k terrorism events.

- Each *role* is filled with one or more entities. There are five such roles for MUC-4: perpetrator individuals (PERPIND), perpetrator organizations (PERPORG), physical targets (TARGET), victims (VICTIM) and weapons (WEAPON). These event roles represent the agents, patients, and instruments associated with terrorism events (Huang and Riloff, 2012).

- Each *role-filler entity* is denoted by a single descriptive *mention*, a span of text from the input document. Because multiple such mentions for each entity may appear in the input, the gold-standard template lists all alternatives (shown in Figure 1), but systems are required to produce just one.

---

[3]The Message Understanding Conferences were a series of U.S. government-organized IE evaluations.

[4]Other types of role fillers include normalized dates and times, and categorical "set" fills. We do not attempt to handle these in the current work.

Evaluation Metric The metric for past work on document-level role-filler *mentions* extraction (Patwardhan and Riloff, 2009; Huang and Riloff, 2011; Du and Cardie, 2020) calculates mention-level precision across all alternative mentions for each role-filler entity. Thus it is not suited for our problem setting, where entity-level precision is needed, where spurious entity extractions will get punished (e.g., recognizing "telephone company building" and "telephone company offices" as two entities will result in lower precision).

Drawing insights from the entity-based CEAF metric (Luo, 2005) from the coreference resolution literature, we design a metric (**CEAF-REE**) for measuring models' performance on this document-level **role-filler entity extraction** task. It is based on maximum bipartite matching algorithm (Kuhn, 1955; Munkres, 1957). The general idea is that, for each role, the metric is computed by aligning gold and predicted entities with the constraint that a predicted (gold) entity is aligned with at most one gold (predicted) entity. Thus, the system that does not recognize the coreferent mentions and use them for separate entities will be penalized in precision score. For the example in Figure 1, if the system extracts "Pilmai telephone company building" and "telephone company offices" as two distinct TARGETs, the precision will drop. We include more details for our CEAF-TF metric in the appendix.

4   REE as Sequence Generation

We treat document-level REE as a sequence-to-sequence task (Sutskever et al., 2014) in order to better model the cross-role dependencies and cross-sentence noun phrase coreference structure. We first transform the task definition into a source and target sequence.

As shown in Figure 2, the *source sequence* simply consists of the tokens of the original document prepended with a "classification" token (i.e., [CLS] in BERT), and appended with a separator token (i.e., [SEP] in BERT). The *target sequence* is the concatenation of target extractions for each role, separated by the separator token. For each role, the target extraction consists of the first mention's

beginning (b) and end (e) tokens:

$$<S> e_{1_b}^{(1)}, e_{1_e}^{(1)}, ... \text{ [SEP]}$$
$$e_{1_b}^{(2)}, e_{1_e}^{(2)}, ... \text{ [SEP]}$$
$$e_{1_b}^{(3)}, e_{1_e}^{(3)}, e_{2_b}^{(3)}, e_{2_e}^{(3)}, ... \text{ [SEP]}$$
$$...$$

Note that we list the roles in a fixed order for all examples. So for the example used in Figure 2, $e_{1_b}^{(1)}, e_{1_e}^{(1)}$ would be "two" and "men" respectively; and $e_{1_b}^{(3)}, e_{1_e}^{(3)}$ would be "water" and "pipes" respectively. Henceforth, we denote the resulting sequence of source tokens as $x_0, x_1, ..., x_m$ and the sequence of target tokens as $y_0, y_1, ..., y_n$.

5   Model: Generative Role-filler Transformer (GRIT)

Our model is shown in Figure 2. It consists of two parts: the encoder (left) for the source tokens; and the decoder (right) for the target tokens. Instead of using a sequence-to-sequence learning architecture with separate modules (Sutskever et al., 2014; Bahdanau et al., 2015), we use a single pretrained transformer model (Devlin et al., 2019) for both parts, and introduce no additional fine-tuned parameters.

Pointer Embeddings The first change to the model is to ensure that the decoder is aware of where its previous predictions come from in the source document, an approach we call "pointer embeddings". Similar to BERT, the input to the model consists of the sum of token, position and segment embeddings. However, for the position we use the corresponding source token's position. For example, for the word "two", the target tokens would have the identical position embedding of the word "two" in the source document. Interestingly, we do not use any explicit target position embeddings, but instead separate each role with a [SEP] token. Empirically, we find that the model is able to use these separators to learn which role to fill and which mentions have filled previous roles.

Our encoder's embedding layer uses standard BERT embedding layer, which applied to the source document tokens. To denote boundary between source and target tokens, we use sequence A (first sequence) segment embeddings for the source tokens, we use sequence B (second sequence) segment embeddings for the target tokens.
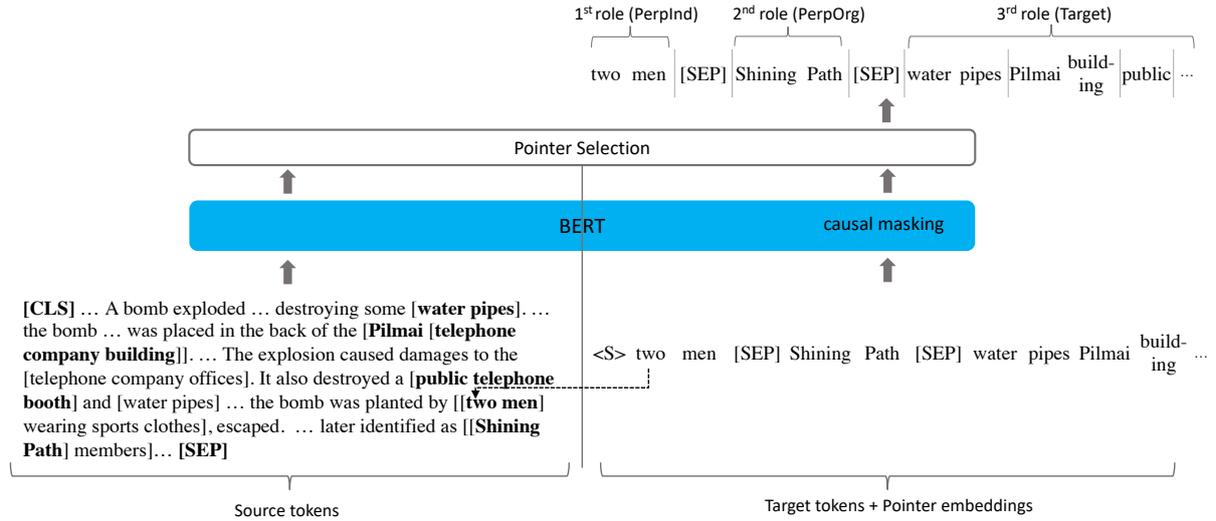
Figure 2: GRIT: generative transformer model for document-level event role-filler entity extraction. (Noun phrase bracketing and **bold** in the source tokens are provided for readability purposes and are not part of the source sequence.)
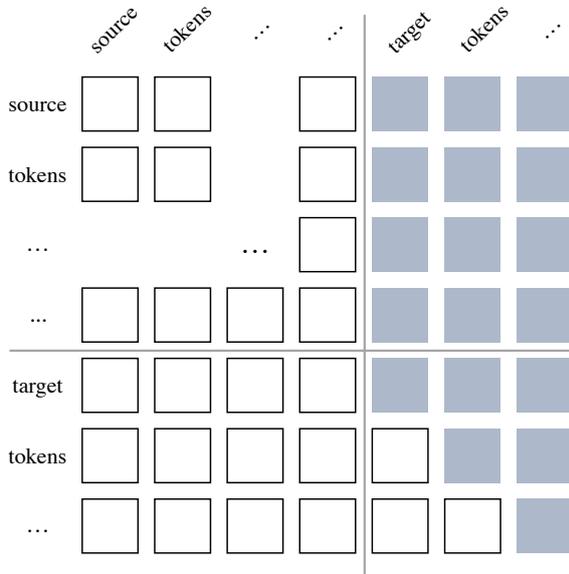

Figure 3: Partially causal masking strategy ($\mathbf{M}$). (White cell: unmasked; Grey cell: masked).

We pass the source document tokens through the encoder's embedding layer, to obtain their embeddings $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_m$. We pass the target tokens $y_0, y_1, ..., y_n$ through the decoder's embedding layer, to obtain $\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_n$.

BERT as Encoder / Decoder We utilize one BERT model as both the source and target embeddings. To distinguish the encoder / decoder representations, we provide a partial causal attention mask on the decoder side.

In Figure 3, we provide an illustration for the attention masks – 2-dimensional matrix denoted as $\mathbf{m}$. For the source tokens, the mask allows full source self-attention, but mask out all target tokens. For $i \in \{0, 1, ..., m\}$,

$$\mathbf{M}_{i,j} = \begin{cases} 1, & \text{if } 0 \le j \le m \\ 0, & \text{otherwise} \end{cases}$$

For the target tokens, to guarantee that the decoder is autoregressive (the current token should not attend to future tokens), we use a causal masking strategy. Assuming we concatenate the target to the source tokens (the joint sequence mentioned below), for $i \in \{m+1, ..., n\}$,

$$\mathbf{M}_{i,j} = \begin{cases} 1, & \text{if } 0 \le j \le m \\ 1, & \text{if } j > m \text{ and } j \le i \\ 0, & \text{otherwise} \end{cases}$$

The joint sequence of source tokens' embeddings $(\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_m)$ and target tokens' embeddings $(\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_n)$ are passed through BERT to obtain their contextualized representations,

$$\begin{aligned} & \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_m, \hat{\mathbf{y}}_0 ..., \hat{\mathbf{y}}_n \\ & = \text{BERT}(\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_m, \mathbf{y}_0, ..., \mathbf{y}_n) \end{aligned}$$

Pointer Decoding For the final layer, we replace word prediction with a simple pointer selection mechanism. For target time step t ($0 \le t \le n$), we first calculate the dot-product between $\hat{\mathbf{y}}_t$ and $\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_m$,

$$z_0, z_1, ..., z_m = \hat{\mathbf{y}}_t \cdot \hat{\mathbf{x}}_0, \hat{\mathbf{y}}_t \cdot \hat{\mathbf{x}}_1, ..., \hat{\mathbf{y}}_t \cdot \hat{\mathbf{x}}_m$$

Then we apply softmax to $z_0, z_1, ..., z_m$ to obtain the probabilities of pointing to each source token,

$$p_0, p_1, ..., p_m = \text{softmax}(z_0, z_1, ..., z_m)$$

Test prediction is done with greedy decoding. At each time step t, argmax is applied to find the source token which has the highest probability. The predicted token is added to the target sequence for the next time step $t + 1$ with its pointer embedding. We stop decoding when the fifth [SEP] token is predicted, which represents the end of extractions for the last role.

In addition, we add the following decoding constraints,

- Tune probability of generating [SEP]. By doing this, we encourage the model to point to other source tokens and thus extract more entities for each role, which will help increase the recall. (We set the hyperparameter of downweigh to 0.01, i.e., for the [SEP] token $p = 0.01 * p$.)

- Ensure that the token position increase from start token to end token. When decoding tokens for each role, we know that mention spans should obey this property. Thus we eliminate those invalid choices during decoding.

6 Experimental Setup

We conduct evaluations on the MUC-4 dataset (1992), and compare to recent competitive end-to-end models (Wadden et al., 2019; Du and Cardie, 2020) in IE (Section 7). Besides the normal evaluation, we are also interested in how well our GRIT model captures coreference linguistic knowledge, and comparison with the prior models. In Section 8, we present relevant evaluations on the subset of test documents.

Dataset and Evaluation Metric The MUC-4 dataset consists of 1,700 documents with associated templates. Similar to (Huang and Riloff, 2012; Du and Cardie, 2020), we use the 1300 documents for training, 200 documents (TST1+TST2) as the development set and 200 documents (TST3+TST4) as the test set. Each document in the dataset contains on average 403.27 tokens, 7.12 paragraphs. In Table 1, we include descriptions for each role in the template.

We use the first appearing mention of the role-filler entity as the training signal (thus do not use the other alternative mentions during training).

Roles	Descriptions
PERPIND	A person responsible for the incident.
PERPORG	An organization responsible for the incident.
TARGET	A thing (inanimate object) that was attacked.
VICTIM	The name of a person who was the obvious or apparent target of the attack or who became a victim of the attack.
WEAPON	A device used by the perpetrator(s) in carrying.

Table 1: Natural Language Descriptions for Each Role.

We use CEAF-REE which is covered in Section 3 as the evaluation metric. The results are reported as Precision (P), Recall (R) and F-measure (F1) score for the micro-average for all the event roles (Table 4). We also report the per-role results to have a fine-grained understanding of the numbers (Table 2).

Baselines We compare to recent strong models for (document-level) information/event extraction. CohesionExtract (Huang and Riloff, 2012) is a bottom-up approach for event extraction that first aggressively identifies candidate role-fillers, and prune the candidates located in event-irrelevant sentences.[5] Du and Cardie (2020) propose *neural sequence tagging (NST)* models with contextualized representations for document-level role filler mentions extraction. We train this model with BIO tagging scheme to identify the first mention for each role-filler entity and its type (i.e., B-PerpInd, I-PerpInd for perpetrator individual). DYGIE++ (Wadden et al., 2019) is a span-enumeration based extraction model for entity, relation, and event extraction. The model (1) enumerates all the possible spans in the document; (2) concatenates the representations of the span's beginning & end token and use it as its representation, and pass it through a classifier layer to predict whether the span represents certain role-filler entity and what the role is. Both the NST and DYGIE++ are end-to-end and fine-tuned BERT (Devlin et al., 2019) contextualized representations with task-specific data. We train them to identify the first mention for each role-filler entity (to ensure fair comparison with our proposed model). Unsupervised event schema induction based approaches (Chambers and Jurafsky, 2011; Chambers, 2013; Cheung et al., 2013) are also able

---

[5]Instead of using feature-engineering based sentence classification to identify event-relevant sentences, we re-implement the sentence classifier with BiLSTM-based neural sequence model.

	PERPIND	PERPORG	TARGET	VICTIM	WEAPON
NST (Du and Cardie, 2020)	48.39 / 32.61 / 38.96	60.00 / 43.90 / 50.70	54.96 / 52.94 / **53.93**	62.50 / 63.16 / 62.83	61.67 / 61.67 / **61.67**
DYGIE++ (Wadden et al., 2019)	59.49 / 34.06 / 43.32	56.00 / 34.15 / 42.42	53.49 / 50.74 / 52.08	60.00 / 66.32 / 63.00	57.14 / 53.33 / 55.17
GRIT	65.48 / 39.86 / **49.55**	66.04 / 42.68 / **51.85**	55.05 / 44.12 / 48.98	76.32 / 61.05 / **67.84**	61.82 / 56.67 / 59.13

Table 2: Per-role performance scored by CEAF-REE (reported as P/R/F1, highest F1 for each role are boldfaced).

	$k = 1$	$1 < k \leq 1.25$	$1.25 < k \leq 1.5$	$1.5 < k \leq 1.75$	$k > 1.75$
NST (Du and Cardie, 2020)	63.83 / 51.72 / 57.14	57.45 / 38.57 / 46.15	60.32 / 49.03 / 54.09	64.81 / 50.00 / 56.45	66.67 / 51.90 / 58.36
DYGIE++ (Wadden et al., 2019)	**72.50** / 50.00 / 59.18	70.00 / 40.00 / 50.91	60.48 / 48.39 / 53.76	52.94 / 38.57 / 44.63	66.96 / 48.73 / 56.41
GRIT	65.85 / 46.55 / 54.55	**74.42** / 45.71 / 56.64	**73.20** / 45.81 / 56.35	**67.44** / 41.43 / 51.33	**69.75** / 52.53 / 59.93

Table 3: Evaluations on the subsets of documents with increasing number of mentions per role-filler entity. k denotes the average # mentions per role-filler entity. Results for each column are reported as Precision / Recall / F1. The highest precisions are boldfaced for each bucket.

to model the coreference relations and entities at document-level, but have been proved to perform substantially worse than supervised models (Patwardhan and Riloff, 2009; Huang and Riloff, 2012). Thus we do not compare with them. We also experimented with a variant of our GRIT model – instead of always pointing to the same [SEP] in the source tokens to finish extracting the role-filler entities for a role, we use five different [SEP] tokens. During decoding, the model points to the corresponding [SEP] as the end of extraction for that role. This variant does not improve over the current best results and we omit reporting its performance.

7 Results

In Table 4, we report the micro-average performance on the test set. We observe that our GRIT model substantially outperforms the baseline extraction models in precision and F1, with an over 5% improvement in precision over DYGIE++.

Table 2 compares the models' performance scores on each role (PERPIND, PERPORG, TARGET, VICTIM, WEAPON). We see that, (1) our model achieves the best precision across the roles; (2) for the roles that come with entities containing more human names (e.g., PERPIND and VICTIM), our model substantially outperforms the baselines; (3) for the role PERPORG, our model scores better precision but lower recall than neural sequence tagging, which results in a slightly better F1 score; (4) for the roles TARGET and WEAPON, our model is more conservative (lower recall) and achieves lower F1. One possibility is that for role like TAR-

Models	P	R	F1
CohesionExtract (Huang and Riloff, 2012)	58.38	39.53	47.14
NST (Du and Cardie, 2020)	56.82	**48.92**	52.58
DYGIE++ (Wadden et al., 2019)	57.04	46.77	51.40
GRIT	**64.19**[**]	47.36	**54.50**[*]

Table 4: Micro-average results (the highest number of each column is boldfaced). Significance is indicated with [**]$(p < 0.01)$,[*]$(p < 0.1)$ – all tests are computed using the paired bootstrap procedure (Berg-Kirkpatrick et al., 2012).

GET, on average there are more entities (though with only one mention each), and it's harder for our model to decode as many TARGET entities correct in a generative way.

8 Discussion

How well do the models capture coreference relations between mentions? We also conduct targeted evaluations on subsets of test documents whose gold extractions come with coreferent mentions. From left to right in Table 3, we report results on the subsets of documents with increasing number (k) of possible (coreferent) mentions per role-filler entity. We find that: (1) On the subset of documents with only one mention for each role-filler entity ($k = 1$), our model has no significant advantage over DYGIE++ and the sequence tagging based model; (2) But as k increases, the advantage of our GRIT substantially increases – with an over 10% gap in precision when $1 < k \leq 1.5$,

and a near 5% gap in precision when $k > 1.5$.

From the qualitative example (document excerpt and the extractions in Figure 4), we also observe our model recognizes the coreference relation between candidate role-filler entity mentions, while the baselines do not, which shows that our model is better at capturing the (non-)coreference relations between role-filler entity mentions. It also proves the advantage of a generative model in this setting.

[P1]... a bomb exploded at the front door of the [home of a peruvian army general], causing damages but no casualties. ... **[P2]** The terrorist attack was ..., by ... who hurled a bomb at the [home of general enrique franco], in the San ... **[P3]** The bomb seriously damaged the [general's [vehicle]], ... and those of [neighboring [houses]].

	TARGET
Gold Role-filler Entities	• home of peruvian army general, home of general enrique franco • vehicle, general's vehicle • houses, neighboring houses
NST	• home of peruvian army general • home of general enrique franco
DYGIE++	• home of peruvian army general • home of general enrique franco • houses
GRIT	• home of peruvian army general • houses

Figure 4: Our model implicitly captures coreference relations between mentions.

How well do models capture dependencies between different roles? To study this phenomenon, we consider nested role-filler entity mentions in the documents. In the example of Figure 1, "shining path" is a role-filler entity mention for PERPORG nested in "two *shining path* members" (a role-filler entity mention for PERPIND). The nesting happens more often between more related roles (e.g., PERPIND and PERPORG) – we find that 33 out of the 200 test documents' gold extractions contain nested role-filler entity mentions between the two roles.

In Table 5, we present the CEAF-REE scores for role PERPORG on the subset of documents with nested roles. As we hypothesized beforehand, GRIT is able to learn the dependency between different roles and can learn to avoid missing relevant role-filler entities for later roles. The results provide empirical evidence: by learning the dependency between PERPIND and PERPORG, GRIT

	PERPORG (all docs)	PERPORG (33/200)
	P / R / F1	P / R / F1
NST	56.00 / 34.15 / 42.42	80.00 / 44.44 / 57.14
DYGIE++	60.00 / **43.90** / 50.70	61.54 / 35.56 / 45.07
GRIT	66.04 / 42.68 / 51.85	80.77 / **46.67** / 59.15

Table 5: Evaluation on the subset of documents that have nested role-filler entity mentions between role PERPIND and PERPORG (highest recalls boldfaced).

...[[[**guerrillas**] of the [**FARC**] and the [**popular liberation army**]] (EPL)] attacked four towns in northern Colombia, leaving 17 guerrillas and 2 soldiers dead and 3 bridges partially destroyed. ...

	PERPIND	PERPORG
Gold Role-filler Entities	• guerrillas, guerrillas of FARC and popular liberation army (EPL)	• EPL, popular liberation army • FARC
NST & DYGIE++	• guerrillas	-
GRIT	• guerrillas	• FARC • popular liberation army

Figure 5: Our model captures dependencies between different roles.

improves the relative recall score on the subset of documents as compared to DYGIE++. On all the 200 test documents, our model is ∼ 2% below DY-GIE++ in recall; while on the 33 docs, our model scores much higher than DYGIE++ in recall.

For the document in the example of Figure 5, our model correctly extracts the two role-filler entities for PERPORG: "FARC" and "popular liberation army", which are closely related to the PERPIND entity "guerrilla". While DYGIE++ and NST both miss the entities for PERPORG.

Decoding Ablation Study In the table below, we present ablation results based on the decoding constraints. These illustrate the influence of the decoding constraints on the our model's performance. The two constraints both significantly improve model predictions. Without downweighing the probability of pointing to [SEP], the precision increases but recall and F1 significantly drops.

	P	R	F1	Δ (F1)
GRIT	64.19	47.36	54.50	
− [SEP] downweigh	67.43	40.12	50.31	-4.19
− constraint on pointer offset	62.90	45.79	53.00	-1.50

Table 6: Decoding Ablation Study

Additional Parameters and Training Cost Finally we consider additional parameters and training time of the models: As we introduced previously, the baseline models DYGIE++ and NST both require an additional classifier layer on top of BERT's hidden state (of size H) for making the predictions. While our GRIT model does not require adding any new parameters. As for the training time, training the DYGIE++ model takes over 10 times longer time than NST and our model. This time comes from the DYGIE++ model requirement of enumerating all possible spans (to a certain length constraint) in the document and calculating the loss with their labels.

	additional params	training cost
DYGIE++	$2H(\#\text{roles} + 1)$	$\sim$20h
NST	$H(2\#\text{roles} + 1)$	$\sim$1h
GRIT	0	<40min

Table 7: Additional Parameters and Training Cost.

9 Conclusion

We revisit the classic and challenging problem of document-level role-filler entity extraction (REE), and find that there is still room for improvement. We introduce an effective end-to-end transformer based generative model, which learns the document representation and encodes the dependency between role-filler entities and between event roles. It outperforms the baselines on the task and better captures the coreference linguistic phenomena. In the future, it would be interesting to investigate how to enable the model to also do template recognition.

Acknowledgments

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Seattle, Washington, USA. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 837–846, Atlanta, Georgia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Xinya Du and Claire Cardie. 2020. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve

event detection via recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 352–361, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. A language-independent neural network for event detection. *Science China Information Sciences*, 61(9):092106.

Ralph Grishman and Beth Sundheim. 1996. Design of the MUC-6 evaluation. In *TIPSTER TEXT PROGRAM PHASE II: Proceedings of a Workshop held at Vienna, Virginia, May 6-8, 1996*, pages 413–422, Vienna, Virginia, USA. Association for Computational Linguistics.

Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *Advances in Neural Information Processing Systems*, pages 7944–7954.

Ruihong Huang and Ellen Riloff. 2011. Peeling back the layers: Detecting event role fillers in secondary contexts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1137–1147, Portland, Oregon, USA. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2012. Modeling textual cohesion for event extraction. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multiscale representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3693–3704, Minneapolis, Minnesota. Association for Computational Linguistics.

Dan Jurafsky and James H. Martin. 2014. *Speech and language processing*. Prentice Hall, Pearson Education International.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Xiang Li, Thien Huu Nguyen, Kai Cao, and Ralph Grishman. 2015. Improving event detection with abstract meaning representation. In *Proceedings of the First Workshop on Computing News Storylines*, pages 11–15, Beijing, China. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

(LDC) Linguistic Data Consortium. 2005. English annotation guidelines for events. *https://www.ldc.upenn.edu/sites/ www.ldc.upenn.edu/files/ english-events-guidelines-v5.4.3.pdf*.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, and Yue Zhang. 2019. Open domain event extraction using neural latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871, Florence, Italy. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

MUC-4. 1992. Fourth message understanding conference (MUC-4). In *Proceedings of FOURTH MESSAGE UNDERSTANDING CONFERENCE (MUC-4)*, McLean, Virginia.

James Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 365–371, Beijing, China. Association for Computational Linguistics.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 151–160, Singapore. Association for Computational Linguistics.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Beth M. Sundheim. 1991. Overview of the third Message Understanding Evaluation and Conference. In *Third Message Uunderstanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*.

Beth M. Sundheim. 1993. The Message Understanding Conferences. In *TIPSTER TEXT PROGRAM: PHASE I: Proceedings of a Workshop held at Fredricksburg, Virginia, September 19-23, 1993*, pages 5–5, Fredricksburg, Virginia, USA. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Hai Wang and Hoifung Poon. 2018. Deep probabilistic logic: A unifying framework for indirect supervision. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1891–1902, Brussels, Belgium. Association for Computational Linguistics.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, 1(2):99–120.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 414–419, Melbourne, Australia. Association for Computational Linguistics.

A Appendices

A.1 CEAF-REE metric

Notations First we provide the necessary notations. Let reference (gold) role-filler entities of one role in a document d be:

$$R(d) = \{R_i : i = 1, 2, ..., |R(d)|\}$$

and predicted role-filler entities be:

$$S(d) = \{S_i : i = 1, 2, ..., |S(d)|\}$$

Let m be the smaller one of $|R(d)|$ and $|S(d)|$, i.e., $m = min(|R(d)|, |S(d)|)$. Let $R_m \subset R$ and $S_m \subset S$ be any subsets with m entities. Let $G(R_m, S_m)$ be the set of *one-to-one* entity maps from R_m to S_m, and G_m be the set of all possible one-to-one maps (of size-m) between subsets of R and S. Obviously, we have $G(R_m, S_m) \in G_m$.

The similarity function $\phi(r, s)$ measures the "similarity" between two entities. It takes nonnegative values: zero-value means role-filler entity r is not subset of s.

$$\phi(r, s) = \begin{cases} 1, & \text{if } s \subseteq r \\ 0, & \text{otherwise} \end{cases}$$

Calculating CEAF-REE score Next we present how to calculate the CEAF-REE score. Given the document d, for a certain event role (e.g., TARGET), with its gold entities R and system predicted entities S, we first find the best alignment g^* by maximizing the total similarity Φ (maximum bipartite matching algorithm is applied in this step):

$$g^* = \arg \max_{g \in G_m} \Phi(g) = \arg \max_{g \in G_m} \sum_{r \in R_m} \phi(r, g(r))$$

Let R_m^* and $S_m^* = g^*(R_m^*)$ denote the gold and predicted role-filler entity subset (respectively), where best matching g^* is obtained. Then the maximum total similarity is,

$$\Phi(g^*) = \sum_{r \in R_m^*} \phi(r, g^*(r))$$

We can also calculate the entity self-similarity with ϕ. Finally, we calculate the precision, recall and F-measure for CEAF-REE as follows:

$$prec = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)}$$

$$recall = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)}$$

$$F = \frac{2 * prec * recall}{prec + recall}$$

Figure 6: Test cases for the CEAF-REE metric (The lines between entities denote the best alignment/matching).

	Precision	Recall	F1
case 1	0.75	1.00	0.86
case 2	1.00	1.00	1.00
case 3	1.00	0.67	0.80

Table 8: CEAF-REE scores for case-$\{1, 2, 3\}$.

We list several cases (Figure 6) and their CEAF-REE scores (Table 8) to facilitate understanding.

For more details, readers can refer to Section 2

of Luo (2005).

A.2 Others

Code and Computing   We use the NVIDIA TITAN Xp GPU for our computing infrastructure. We build our model based on the Hugging-face NER models' implementation `https://github.com/huggingface/transformers/tree/3ee431dd4c720e67e35a449b453d3dc2b15ccfff/examples/ner`. The hyperparameters can also be obtained from the default values in the repo.

Link to Corpus   The raw corpus and preprocessing script can be found at: `https://github.com/brendano/muc4_proc`

Dependencies

- Python 3.6.10

- Transformers: transformers 2.4.1 installed from source.

- Pytorch-Struct: Install from Github.

- Pytorch-Lightning==0.7.1

- Pytorch==1.4.0