Open Relation and Event Type Discovery with Type Abstraction

Sha Li, Heng Ji, Jiawei Han
University of Illinois Urbana-Champaign
{shal2, hengji, hanj}@illinois.edu

Abstract

Conventional "closed-world" information extraction (IE) approaches rely on human ontologies to define the scope for extraction. As a result, such approaches fall short when applied to new domains. This calls for systems that can automatically infer new types from given corpora, a task which we refer to as *type discovery*. To tackle this problem, we introduce the idea of type abstraction, where the model is prompted to generalize and name the type. Then we use the similarity between inferred names to induce clusters. Observing that this abstraction-based representation is often complementary to the entity/trigger token representation, we set up these two representations as two views and design our model as a co-training framework. Our experiments on multiple relation extraction and event extraction datasets consistently show the advantage of our type abstraction approach.

1 Introduction

Information extraction has enjoyed widespread success, however, the majority of information extraction methods are "reactive", relying on end-users to specify their information needs in prior and provide supervision accordingly. This leads to "closed-world" systems (Lin et al., 2020; Du and Cardie, 2020; Li et al., 2021; Zhong and Chen, 2021; Ye et al., 2022) that are confined to a set of pre-defined types. It is desirable to make systems act more "proactively" like humans who are always on the lookout for interesting new information, generalize them into new types, and find more instances of such types, even if they are not seen previously.

One related attempt is the Open Information Extraction paradigm (Banko et al., 2008), which aims at extracting all (subject, predicate, object) triples from text that denote some kind of relation. While OpenIE does not rely on pre-specified relations, its exhaustive and free-form nature often leads to noisy and redundant extractions.

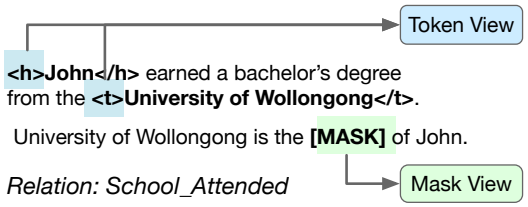

Figure 1: For each instance, the token view is computed from the pre-trained LM embedding of the first token in entity/trigger. The mask view is computed from the [MASK] token embedding in the type prompt.

To bridge the gap between closed-world IE and OpenIE, a vital step is for systems to possess the ability of automatically inducing new types and extracting instances of such new types. Under various contexts, related methods have been proposed under the name of "relation discovery" (Yao et al., 2011; Marcheggiani and Titov, 2016),"open relation extraction" (Wu et al., 2019; Hu et al., 2020) and "event type induction" (Huang and Ji, 2020; Shen et al., 2021). In this paper, we unify such terms and refer to the task as *type discovery*.

Type discovery can naturally be posed as a clustering task. This heavily relies on defining an appropriate metric space where types are easily separable. The token embedding space from pre-trained language models is a popular choice, but as observed by (Zhao et al., 2021), the original metric space derived from BERT (Devlin et al., 2019) is often prone to reflect surface form similarity rather than the desired relation/event-centered similarity. One way to alleviate this issue is to use known types to help learn a similarity metric that can also be applied to unknown types (Wu et al., 2019; Zhao et al., 2021; Huang and Ji, 2020).

In this paper we introduce another idea of *abstraction*: a discovered type should have an appropriate and concise type name. The human vocabulary serves as a good repository of concepts that appear meaningful to people. When we assign a name to a cluster, we implicitly define the com-

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6864 - 6877
December 7-11, 2022

Relation	Mask view	Token view	Δ
website (of org)	0.2424	0.9366	-0.6941
age (of person)	0.2896	0.389	-0.0994
founded_by	0.2734	0.1268	0.1466
employee_of	0.4434	0.2703	0.1731
Avg	0.3678	0.2989	0.0688

Table 1: Probing k-NN Accuracy of the token view and the mask view on distinguishing relations without training. We compute k-NN using cosine similarity of embeddings with $k = 32$ on TACRED (Zhang et al., 2017). While on average the mask view outperforms the token view, the two views excel at different types.

monality of instances within the cluster and also the criteria for including new instances to the cluster. Since masked language models have the ability to "fill in the blank", with the help of a *type-inducing prompt* as shown in Figure 1, we can guide the model to predict a name or indicative word for any relation/event instance. Moreover, since inferring the best name for a cluster from a single instance is a difficult task, we do not require this prediction to be exact: we utilize the similarity between predicted names to perform clustering.

This abstraction-based representation is complementary to the widely-adopted token-based representation of relations/events. We refer to our abstraction-based representation as "mask view" since the embedding for the instance is derived from the [MASK] token. Alternatively, we can also compute a "token view" derived from the pre-trained LM embeddings of the involved entity/trigger directly. As shown in Table 1, without any training, the token-based representation (token view) and the type abstraction representation (mask view) specialize in different types. When the relation type is strongly connected to the entity type as in "website", the token view provides a strong prior. The mask view can distinguish relations with similar entity types (person, organization) based on relational phrases such as "found, create, work at".

Therefore, we combine the mask view and the token view in a co-training framework (Blum and Mitchell, 1998), utilizing information from both ends. As shown in Figure 2, our model consists of a shared contextual encoder, two view-specific projection networks and classification layers for known and unknown types respectively. Since no annotation is available for new types, we perform clustering over the two views to obtain pseudo-labels and then use such labels to guide the training of the classification layer of the opposite view.

We apply our model to both relation discovery and event discovery with minimal changes to the type-inducing prompt. Our model serves the dual proposes of (1) inducing clusters with exemplar instances from the input corpus to assist ontology construction and (2) serving as a classifier for instances of unknown types. On the task of relation discovery our model outperforms the previous transfer-learning based SOTA model by 4.3% and 2.2% accuracy on benchmark datasets TACRED and FewRel respectively. On event discovery we also set the new SOTA, achieving 77.3% accuracy for type discovery with gold-standard triggers.

The main contributions of this paper include:

- We propose the idea of type abstraction, implicitly using inferred type names from the language model to improve type discovery.
- We design a co-training framework that combines the advantage of type abstraction and the conventional token-based representation.
- We show that our model can be applied to the discovery of both relation types and event types and achieve superior performance over existing models on both tasks.

2 Problem Definition

We first define the task of **type discovery** and then discuss the realization of this task to relations and events.

Given a set of unlabeled instances $D^u = \{x_1^u, x_2^u, \cdots, x_M^u\}$ and an estimated number of unknown types $|C^u|$, the goal of type discovery is to learn a model f that can map $x \in X^u$ into one of $y \in C^u$ unknown types.

In the case of relation discovery, each instance x is an entity mention pair $\{h, t\}$ embedded within a sentence context. As shown in Figure 1, the instance is "**John** earned a bachelor's degree from the **University of Wollongong**" with the head entity mention "John" and the tail entity mention "University of Wollongong". Each entity mention is a span with start and end indexes (s, e) in the sentence. The associated label y in this case is relation type "School_Attended".[1]

In the case of event discovery, each instance x is a trigger word/phrase mention t with start and

[1]In the relation extraction literature, the relation type is often denoted as r. For unified notation we use y.

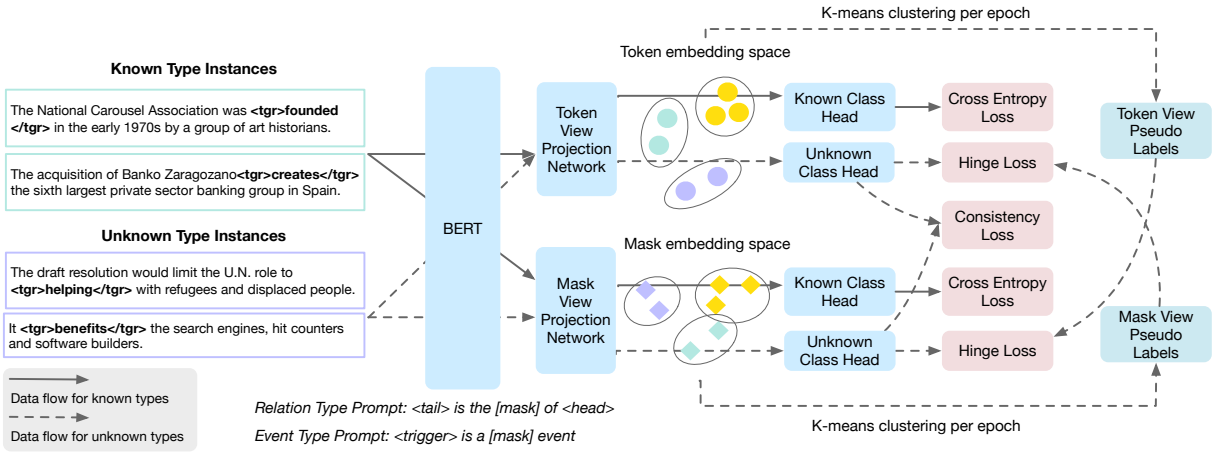

Figure 2: Overview of our type discovery model. We show two instances of the `Start-Org` event type (green) and two instances of `Assist` event type (purple). For each instance, we compute the token view and the mask view through two separate projection networks. We use K-means clustering in the respective embedding spaces to obtain pseudo labels and use the labels to supervise the alternative view.

end indexes (s, e) in a sentence context as shown in Figure 2. The label y is the event type. Note that for both relations and events, it is possible for multiple instances to appear within the same sentence, but they have different entity or trigger mentions.

To assist the learning of such a model, we further assume that we have access to a set of labeled instances $D^l = \{(x_1^l, y_1^l), (x_2^l, y_2^l), \cdots, (x_N^l, y_N^l)\}$. The type labels $Y = \{y_1^l, y_2^l, \cdots, y_N^l\}$ present in D^l belong to C^l known classes which are disjoint from the classes to discover, namely $C^l \cap C^u = \emptyset$.

3 Method

Our model is built on the observation that the token view and the mask view are often complementary and work well for different types. Thus, the core of our model is the construction of two views and how they can be utilized for co-training.

3.1 Instance Representation

We first describe how the relation instances are represented and then discuss the changes for event instances. Similar to (Baldini Soares et al., 2019), in the input sentence we mark up the entity/trigger with special tokens. We use $\langle h \rangle$ and $\langle t \rangle$ for head and tail entities respectively and $\langle tgr \rangle$ for the trigger. For each instance we have two views: the token view and the mask view. The two views share the same BERT (Devlin et al., 2019) encoder, but have slightly different inputs.

Relation Instances. For the token view, we embed the sentence using BERT and take the embedding for the first token in the entity (index s) as the entity representation.[2] We concatenate the representations for the head and tail entity to obtain the relation representation (Baldini Soares et al., 2019).

$$\begin{aligned} \vec{h} &= \text{BERT}(x)[s_h]; \vec{t} = \text{BERT}(x)[s_t] \\ \vec{x}_1 &= [\vec{h}; \vec{t}] \end{aligned} \tag{1}$$

For the mask view, we append a *type prompt* p_r to the input sentence. The type prompt is designed so that the relation type name should be fit into the `[MASK]` token position. For relations, we use the prompt of "$\langle$tail$\rangle$ is the `[MASK]` of $\langle$ head $\rangle$ " where $\langle$tail$\rangle$ and $\langle$ head $\rangle$ are replaced by the actual head and tail entity strings for each instance. Then we embed the sentence along with the type prompt with BERT and use the embedding for the `[MASK]` token as the relation representation.

$$\vec{x_2} = \text{BERT}(x; p_r)[s_{mask}] \tag{2}$$

Event Instances. For event instances in the token view we use the embedding for the first token in the trigger mention as the event representation. In the mask view we use a different type prompt p_e: "$\langle$ trigger $\rangle$ is a `[MASK]` event" where $\langle$ trigger $\rangle$ is replaced by the actual trigger.

$$\begin{aligned} \vec{x_1} &= \text{BERT}(x)[s] \\ \vec{x_2} &= \text{BERT}(x; p_e)[s_{mask}] \end{aligned} \tag{3}$$

3.2 Multi-view Model

Our model consists of a shared BERT encoder, two projection networks f and four classifier heads g

[2] We overload the notation a bit here and use x to denote the sentence where the instance is from.

(for known types and unknown types per view, respectively).

The projection networks map the instance representation $\vec{x}$ to a lower dimension space representation $\vec{h}$ and the classifier heads g maps $\vec{h}$ into logits $\vec{l}$ corresponding to the labels.

$$\begin{aligned} \vec{h} &= f(\vec{x}) \\ \vec{l^u} &= g^u(\vec{h}); \vec{l^l} = g^l(\vec{h}) \\ \vec{\hat{y}} &= \text{softmax}\left([\vec{l^l}; \vec{l^u}]\right) \end{aligned} \tag{4}$$

For instances of known classes, we use the cross-entropy loss with label smoothing to train the network:

$$\mathcal{L}^l = -\frac{1}{|D^l|} \sum_{D^l} \sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{5}$$

For instances of unknown classes, we run K-means clustering on the projection network output to assign pseudo-labels:

$$\tilde{y}^u = \text{K-means}(\vec{h}) \in \{1, \cdots, C^u\} \tag{6}$$

As the pseudo-label assignment might not align across views (cluster 1 in the token view is not the same as cluster 1 in the mask view), for each batch of instances, we further transform the cluster assignment labels into pairwise labels $q_{ij} = \mathbb{1}(\tilde{y}_i = \tilde{y}_j)$.

We compute the discrepancy between the predictions of the pair x_i, x_j using the Jensen-Shannon(JS) divergence:

$$d_{ij} = \frac{1}{2}\left\{\text{KL}(\hat{\vec{y}}_i || \hat{\vec{y}}_j) + \text{KL}(\hat{\vec{y}}_j || \hat{\vec{y}}_i)\right\} \tag{7}$$

Then the loss function for an unlabeled pair is defined as the JS divergence if two instances are assigned to the same cluster and a hinge loss over the JS divergence if two instances are assigned to different clusters.

$$\begin{aligned} l(d_{ij}, q_{ij}) &= q_{ij} d_{ij} + (1 - q_{ij}) \max(0, \alpha - d_{ij}) \\ \mathcal{L}^u &= \frac{1}{\binom{|D^u|}{2}} \sum_{x_i, x_j \in D^u} \left(l(d^1_{ij}, q^2_{ij}) + l(d^2_{ij}, q^1_{ij})\right) \end{aligned} \tag{8}$$

where d^1_{ij} is computed from the token view, d^2_{ij} is computed from the mask view and the similarly for q^1_{ij} and q^2_{ij}. α is a hyper-parameter for the hinge loss.

If a single view was used, this loss falls back to the contrastive loss term defined for unlabeled instances in (Hsu et al., 2018; Zhao et al., 2021).

Dataset	Known		Unknown	
	#Classes	# Ins	#Classes	#Ins
TACRED	31	23,477	10	1,996
FewRel	64	44,800	16	11,200
ACE-controlled	10	4,089	23	1,221
ACE-end2end	10	1,663	-	17,172

Table 2: Statistics of the datasets used. The first two are for relation discovery and the last two datasets are used for event discovery.

In the training process, we observe that since the pseudo label $\tilde{y}$ is used as the target for the opposite view, when these two views produce very different clusters, it leads to performance oscillation over epochs.

To alleviate this issue, we add a consistency loss that encourages the predictions of the two views to be similar to each other:

$$\mathcal{L}^c = \frac{1}{|D^u|} \sum_{D^u} \text{JSD}(\hat{y}^1, \hat{y}^2) \tag{9}$$

The final loss function is a weighted sum of the aforementioned terms:

$$\mathcal{L} = \mathcal{L}^l + \mathcal{L}^u + \beta \mathcal{L}^c \tag{10}$$

β is a hyperparameter and empirically set to 0.2 in our experiments.

3.3 Training Procedure

Before we train our model with the loss function in Equation 10, we warmup our model by pre-training on the labeled data. The loss function here is simply the cross-entropy loss $\mathcal{L}_{pre} = \mathcal{L}^l$.

After pre-training, we load the weights for BERT and the projection networks f to the model for further training. Note that we do not keep the weights for the known class classifier head g^l.

4 Experiments

In the following experiments, we refer to our model as TABS to stand for "type abstraction".

4.1 Relation Discovery Setting

Datasets. We follow RoCORE (Zhao et al., 2021) and evaluate our model on two relation extraction benchmark datasets: TACRED (Zhang et al., 2017) and FewRel (Han et al., 2018). For the TACRED dataset, 31 relation types are treated as known and 10 relation types are unknown, with the types defined in the TAC-KBP slot filling task (Ji and Gr-

ishman, 2011) [3]. Instances with the no_relation label are filtered out as in (Zhao et al., 2021). For the FewRel dataset, we treat the 64 relation types in the original training set as known relations and 16 relation types in the original development set as unknown relations. For both datasets, we leave out 15% of the instances of both known and unknown relation types, and we report results on the set of unknown relation instances. [4]

Baselines. We primarily compare with RoCORE (Zhao et al., 2021) and RSN (Wu et al., 2019). which is the state-of-the-art for relation discovery. RoCORE earns its name from their proposed "relation-oriented clustering module" that attempts to shape the latent space for clustering by a center loss (which pushes instances towards centroids) and a reconstruction loss. We also compare with RSN (Wu et al., 2019), which learns a pairwise similarity metric between relations and transfers such a metric to unknown instances. The encoder is replaced by BERT (Devlin et al., 2019) for a fair comparison. RSN originally uses the Louvain algorithm (Blondel et al., 2008) for clustering, however we observe that sometimes this would lead to all instances assigned to the same cluster so we experiment with a variant using spectral clustering that takes the same graph input as Louvain. For the Louvain variant, we report the best run instead of average and deviation due to cases of clustering collapse.

4.2 Event Discovery Setting

Datasets. We use ACE under the processing by (Lin et al., 2020) for our event discovery experiments. We follow (Huang and Ji, 2020) and set the 10 most popular event types as known types and the remaining 23 event types to be discovered. As ACE is of relatively smaller size compared to the previous datasets used for relation discovery, we leave out 30% of the instances for testing. Results are reported for the unknown type instances only.

Controlled Setting. In the controlled setting we give the models access to ground truth trigger mentions. We compare with the SS-VQ-VAE model from (Huang and Ji, 2020) and the spherical latent clustering model from (Shen et al., 2021). As the two models originally operated on a different set of instances (sense-tagged triggers in (Huang and Ji, 2020) and predicate-object pairs in (Shen et al., 2021)), we reimplement these methods to work with the gold-standard trigger mentions from ACE.

End-to-end Setting. In the end-to-end setting for our system we treat all non-auxiliary verbs as candidate trigger mentions. For the 10 known types, if the annotated trigger matches with one of the candidate trigger mentions, we treat that instance as labeled. All remaining candidate triggers are treated as unknown and we set the number of unknown types $K = 100$. Under this setting, we compare with the full pipeline of ETypeClus (Shen et al., 2021).

4.3 Metrics

The following metrics for cluster quality evaluation are adopted: **Accuracy**, **BCubed-F1** (Bagga and Baldwin, 1998), **V measure** (Rosenberg and Hirschberg, 2007), **Adjusted Rand Index(ARI)** (Hubert and Arabie, 1985).[5] **Accuracy** is computed by finding the maximal matching between the predicted clusters and the ground truth clusters using the Jonker-Volgenant algorithm (Crouse, 2016)[6].

4.4 Implementation Details

We use `bert-base-uncased` as our base encoder. The projection network f is implemented as a two layer MLP with dimensions 768-256-256 and ReLU activation. The classifier heads are implemented as two layer MLPs as well, with dimensions of 256-256-C, where C is either the number of known types or unknown types. For additional hyperparameters, see Appendix Section A.

4.5 Main Results

We present results on relation discovery in Table 3. While all models benefit from transferring relation knowledge from known types to unknown types, RSN (Wu et al., 2019) separates the clustering step from the representation step, so the representations are not highly optimized for clustering unlike RoCORE (Zhao et al., 2021) and our model.

[3] The unknown relations are `schools_attended`, `cause_of_death`, `city_of_death`, `stateorprovince_of_death`, `founded`, `country_of_birth`, `date_of_birth`, `city_of_birth`, `charges`, `country_of_death`.

[4] As the data split is random, our reported numbers are not exactly the same.

[5] The implementation of BCubed is from https://github.com/m-wiesner/BCUBED, and the implementation of V measure and ARI are from the `sklearn` library.

[6] Implementation from https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

Dataset	Model	Acc	B^3 F1	V measure	ARI
TACRED	RSN (Wu et al., 2019)	0.7645 ± 0.034	0.7194 ± 0.036	0.7587 ± 0.030	0.6456 ± 0.047
	RSN-spectral	0.7425 ± 0.041	0.7163 ± 0.013	0.7569 ± 0.013	0.635 ± 0.047
	ROCORE (Zhao et al., 2021)	0.8468 ± 0.059	0.8307 ± 0.031	0.8612 ± 0.019	0.7867 ± 0.052
	TABS	**0.8896** ± 0.011	**0.8535** ± 0.016	**0.8718** ± 0.017	**0.8276** ± 0.018
FewRel	RSN (Wu et al., 2019)	0.4880	0.4783	0.6718	0.4184
	RSN-Spectral	0.6277 ± 0.021	0.6306 ± 0.030	0.7351 ± 0.020	0.5490 ± 0.030
	ROCORE (Zhao et al., 2021)	0.7801 ± 0.012	**0.7652** ± 0.025	**0.8407** ± 0.016	0.7039 ± 0.022
	TABS	**0.8022** ± 0.023	0.7606 ± 0.026	0.8374 ± 0.018	**0.7266** ± 0.032

Table 3: Relation discovery results on TACRED and FewRel. Experiments are ran with 5 different seeds and we report the average score and standard deviation.

Model	**Acc**	**B^3 F1**	**V measure**	**ARI**
		Controlled Setting		
Spherical Clustering (Shen et al., 2021)	0.3830	0.3861	0.5470	0.2726
SS-VQ-VAE (Huang and Ji, 2020)	0.2951	0.2921	0.4063	0.1242
TABS	**0.7732** ± 0.023	**0.7110** ± 0.034	**0.8028**± 0.027	**0.6647** ± 0.038
		End-to-end Setting		
TABS	0.5089	0.5611	0.7049	0.3629

Table 4: Event discovery results on ACE.

Compared with RoCORE, our model (1) employs a multiview representation; (2) removes the relation-oriented clustering module and (3) uses a simpler pretraining procedure with only known classes. Although the training procedure is simplified, the use of both token features and mask features leads to improved effectiveness of the model.

On the event type discovery task in Table 4, we show that our model has a great advantage over unsupervised methods such as spherical latent clustering model (Shen et al., 2021) that does not make use of known types. Among models that perform transfer learning, SS-VQ-VAE (Huang and Ji, 2020) does not employ a strong clustering objective over the unknown classes. In the end-to-end setting, our model still outperforms the previous work. However, the gap between the end-to-end performance and the controlled performance show that extra processing on trigger might be necessary before apply this model to the wild. In the human evaluation Table 5, annotators judged 70% of discovered clusters to be semantically coherent compared to 59% of the clusters from the ETypeClus pipeline.

Model	Cluster Coherence Rate	Instance Discernability Rate
ETypeClus	0.59	0.682
TABS	**0.70**	**0.725**

Table 5: Human evaluation for end-to-end event discovery on the cluster level and instance level. Reported numbers are the ratio of clusters/instances rated as "coherent"/"discernable" by annotators. Cohen's $\kappa = 0.426$ for this binary decision process. More details about the evaluation protocol can be found in Appendix Section C.

Model	Acc	B^3	PL Acc
Full model	**0.903**	**0.881**	**0.878**
A:Token view only	0.849	0.828	0.820
B:Mask view only	0.849	0.832	0.822
C:Two branch token	0.866	0.843	0.833
D:Two branch mask	0.869	0.844	0.837

Table 6: Comparison with model variations on TACRED. PL Acc is the pseudo label accuracy computed from K-means. (Results are from a single run with the same random seed.)

4.6 Ablation Study

Different Views We compare our full model with several ablations of obtaining the different views as shown in Table 6. Variants A and B use only one view to represent the instance showing the advantage of co-training. We further experiment with different ways of constructing the two views. Variant C first computes the token representation of the instance and then apply two different dropout functions to construct two views. This dropout operation can serve as task-agnostic data augmentation, which has proved to be effective for representation learning (Gao et al., 2021). Variant D uses

Figure 3: Comparison of predicted relation clusters on TACRED. Instances in the purple box are shared, instances in the pink box are unique to ROCORE output and instances in the blue box are unique to our model's output.

Model	Acc	B^3
Full Model	**0.903**	**0.881**
w/o supervised pre-training	0.856	0.862
w/o consistency loss	0.896	0.868

Table 7: Ablation studies on the training process for TACRED.

two different type abstraction prompts to construct two representations for the same instance. Both of these variants are more effective than the single view variants but not as effective as combining the token view and the mask view.

Model Design In Table 7 we compare the performance of our full model with variants that omit the pre-training stage and the consistency loss. Pre-training the model on known types is critical to the model's final performance. The consistency loss, while useful, does not contribute as much to the accuracy but rather alleviates the model oscillation over epochs.

Clustering Method In Table 9 we experiment with different clustering methods under our framework. All implementations are from the `sklearn` library. For the spectral clustering variant, we use the default radial basis function (RBF) kernel to compute the affinity matrix[7], whereas for the other clustering methods we using Eulidean distance to compute the affinity matrix. This metric difference might explain why spectral clustering is underperforming. While DBSCAN and Agglomerative-Ward both achieve reasonably good performance, we observe that DBSCAN is quite sensitive to its `eps` parameter, which defines the maximum distance between two samples for one to be considered in the neighborhood of the other. In fact, this parameter needs to be set differently for different random seeds based on the distribution of the nearest neighbor distance. In general, k-means clustering is both stable and efficient for our use case. Note that both our model and ROCORE use k-means clustering to obtain pseudo labels.

[7]This is different from the spectral clustering variant of RSN, where the graph is precomputed following (Wu et al., 2019).

5 Analysis

Predicted Type Names. In Table 8 we show the predicted type names produced by our model. Although our model does not directly rely on such names (but rather the similarity of `[MASK]` embeddings) to perform clustering , the predictions give insights into the internal workings of the model. For example, the predicted names for the `per:cause_of_death` cluster are strongly related to disease. In contrast, the following instance ***Assaf Ramon**, 21, died on Sunday when the F-16 fighter **jet he was flying crashed*** was abstracted to names such as *death, rotor, life, loss* and as a result, was not included as part of the cluster.

Relation Discovery. In Figure 3 we examine the differences in the predicted relation clusters. In the first relation `city_of_death`, ROCORE incorrectly merges many instances of `city_of_birth` into the target cluster. These two relation types not only share the same entity types of (person, city) but can also involve the exact same entities, e.g. Jerusalem. As ROCORE is primary relying on token features to make the prediction, instances with shared entities have high similarity and this propagates errors to other instances. In the second example, we observe that both models work well in more conventional cases, but when it comes to rare values such as "hunger strike", only our model can correctly identify it as the `cause_of_death`.

Best Matched Type	Predicted Names	Instance
	TACRED	
per:date_of_birth	birthday, year, years, february, month	**McNair**, born on **Dec. 14 , 1923**, in the rural Low Country of South Carolina, ...
per:cause_of_death (by disease)	pmid, *cord*, sign, diagnosed, cause	Palestinian leader **Abu Daoud**, who planned the daring deadly attack died Saturday of **illness** ...
per:charges	felony, *and*, *anything*, cocaine, wrong	Wen Qiang, was also accused of **rape** and being unable to explain the sources of **his** assets ...
	ACE	
Personnel:Start-Position	first, inaugural, introduced, appointed, unopposed	The ruling Millennium Democratic Party (MDP)... has suffered declining popularity since President Roh Moo-Hyun **took office** in February.
Business:Start-Org, End-Org, Merge-Org	separate, fold, new, employee, strategic	Major US insurance group AIG is in a deal to **create** Japan 's sixth largest life insurer
Conflict:Demonstrate	street, demonstration, protest, march, picket	Chalabi staged his own **rally** yesterday to support his bid to become the next leader of Iraq.

Table 8: Predicted type names by our model. The names are sorted by frequency of appearance in top 10 predictions. We skip word pieces (starting with ##). Additionally, we show the top-1 instance according to prediction probability.

Clustering	Acc	B^3
k-means	**0.9030**	**0.8806**
DBSCAN	0.8595	0.8481
Agglomerative-Ward	0.8495	0.8497
Spectral	0.7324	0.7258

Table 9: Comparison of different clustering methods.

Event Discovery. We show predicted clusters of event types from our algorithm under the controlled setting in Table 10. Our model is able to handle (1) *diverse triggers*, e.g. "chosen", "appoint" and "becoming" all refer to the Start-Position event type; (2) *ambiguous triggers* such as "becoming" and "filled" cannot be assigned to the event type without referring to the context; and (3) *multi-word triggers*, e.g. "take into custody" refers to Arrest. In the Start-Position cluster, we see a few mis-classified instances of Nominate. These two event types are similar as they both involve a person and a position/title, the difference being whether the person has already been appointed the position or not.

Remaining Challenges *Abstract types.* Relation types such as "part_of","instance_of" and "same_as" from the FewRel dataset are highly abstract and can be associated with various types of entities. In fact, such relations are often best dealt with separately in the context of hypernym detection (Roller et al., 2018), taxonomy construction (Aly et al., 2019; Huang et al., 2020; Chen et al., 2021) or synonym identification (Fei et al., 2019; Shen et al., 2020).

Misaligned level of granularity. We observe that our automatically induced clusters are sometimes not at the same level of granularity as clusters defined by human annotation. For instance the discovered `per:cause_of_death` cluster is more like `per:disease_of_death` and the several business-related events `Start-Org`, `End-Org` and `Merge-Org` are combined into a single cluster. This calls for models that can produce multi-level types or account for human feedback (the user can specify whether the cluster needs to be further split).

6 Related Work

Relation Type Discovery Early work in this direction represented relations as clusters of lexical patterns or syntactic paths (Hasegawa et al., 2004; Shinyama and Sekine, 2006; Yao et al., 2011, 2012; Min et al., 2012; Lopez de Lacalle and Lapata, 2013). A wave of newer methods used learned relation representations (Marcheggiani and Titov, 2016; Yu et al., 2017; Simon et al., 2019; Wu et al., 2019; Tran et al., 2020; Hu et al., 2020; Liu et al., 2021; Zhao et al., 2021), often defining the relation as a function of the involved entities. One key observation made by RSN (Wu et al., 2019) and RoCORE (Zhao et al., 2021) is the possibility of relational knowledge transfer from known relation types to new types. In this work, we also adopt this transfer setting and introduce a new idea of *abstraction*: a relation cluster is meaningful if it aligns well with a human concept.

Event Type Discovery Our task of event discovery is similar to the verb clustering task in SemEval 2019 (QasemiZadeh et al., 2019) which requires mapping verbs in context to semantic

Matched Type	Instances in Cluster
Personnel: Start-Position	Condi Rice has been **chosen** by President Bush to become the new Secretary of State ... If you were president, which national figures would you **appoint** to your cabinet and why? Al-Douri taught international law at Baghdad University before **becoming** a diplomat ... Chui Sai On, who has been **named** [Personnel: Nominate] director of the SARS task force ...
Conflict:Demonstrate	Some 70 people were arrested Saturday as demonstrators clashed with police at the end of a major peace **rally** ... Between 2,500 and 3,000 people **picketed** the CNN studios in Los Angeles ... The crowd **filled** the street leading to the Kazimiya mosque in the northeast of Baghdad ...
Justice: Arrest-Jail	Some 70 people were **arrested** Saturday as demonstrators clashed with police at the end of a major peace rally ... Ferris disappeared from sight, and CNN has confirmed he was **taken into custody**.

Table 10: Predicted clusters of event instances on ACE. The triggers are marked in bold.

frames.[8] ETypeClus (Shen et al., 2021) represents events as (predicate, object) pairs and design a reconstruction-based clustering method over such P-O pairs. SS-VQ-VAE (Huang and Ji, 2020) leverages a vector quantized variational autoencoder model to utilize known types.

7 Conclusions and Future Work

In this paper we study the *type discovery* problem: automatically identifying and extracting new relation and event types from a given corpus. We propose to leverage *type abstraction*, where the model is prompted to name the type, as an alternative view of the data. We design a co-training framework, and demonstrate that our framework works favorably in both relation and event type discovery settings. Currently we have assumed that the new types are disjoint to the old types and the model operates similarly to a transfer learning setting. While the model can be easily extended to handle both new types and old types, more analysis might be needed in this direction. One potential direction would be to explore a continual learning setting, where new types could emerge periodically.

8 Limitations

In this paper we studied datasets that are English and mostly in the newswire genre. Although our method is not strictly restricted to English, the design of the type-inducing prompt will require some prior knowledge about the target language.

For both relation and event type discovery, the model requires the input of candidate entities pairs or triggers. As shown in Table 4, there is a large gap in model performance between the controlled setting and the end-to-end setting (although this could be partially attributed to incomplete annotation and our simple candidate extraction process). This would limit the model's application in the real world and we believe this should be the focus of future research.

[8] We were not able to follow this setting due to unavailable data.

9 Ethical Considerations

Intended use. The model introduced in this paper is intended to be used for exploratory analysis of datasets. For instance, when presented with a new corpus, the model can be used to extract clusters of new event types that can then be judged by human annotators and used as a basis for developing an event ontology and event extraction system.

Biases. The model does not perform any filtering of its input. If the input corpus contains mentions of discriminatory or offensive language, the model will be unaware and will likely surface such issues in its output.

Acknowledgements

This research was supported by US DARPA KAIROS Program No. FA8750-19-2-1004. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

References

Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko. 2019. Every child should have parents: A taxonomy refinement algorithm based on hyperbolic term embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4811–4817, Florence, Italy. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2008. Open information extraction from the web. In *CACM*.

Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:10008.

Avrim Blum and Tom. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT' 98*.

Catherine Chen, Kevin Lin, and Dan Klein. 2021. Constructing taxonomies from pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4687–4700, Online. Association for Computational Linguistics.

David Frederic Crouse. 2016. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52:1679–1696.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Hongliang Fei, Shulong Tan, and Ping Li. 2019. Hierarchical multi-task word embedding learning for synonym prediction. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, Barcelona, Spain.

Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. 2018. Learning to cluster in order to transfer across domains and tasks. *ICLR*.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip Yu. 2020. SelfORE: Self-supervised relational feature learning for open relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3673–3682, Online. Association for Computational Linguistics.

Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang, and Jiawei Han. 2020. Corel: Seed-guided topical taxonomy construction by concept learning and relation transferring. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724, Online. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proc. ACL2011*.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 7999–8009, Online. Association for Computational Linguistics.

Fangchao Liu, Lingyong Yan, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Element intervention for open relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4683–4693, Online. Association for Computational Linguistics.

Oier Lopez de Lacalle and Mirella Lapata. 2013. Unsupervised relation extraction with general domain knowledge. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 415–425, Seattle, Washington, USA. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2016. Discrete-state variational autoencoders for joint discovery and factorization of relations. *Transactions of the Association for Computational Linguistics*, 4:231–244.

Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1027–1037, Jeju Island, Korea. Association for Computational Linguistics.

Behrang QasemiZadeh, Miriam R. L. Petruck, Regina Stodden, Laura Kallmeyer, and Marie Candito. 2019. SemEval-2019 task 2: Unsupervised lexical frame induction. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 16–30, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363, Melbourne, Australia. Association for Computational Linguistics.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic. Association for Computational Linguistics.

Jiaming Shen, Wenda Qiu, Jingbo Shang, Michelle Vanni, Xiang Ren, and Jiawei Han. 2020. SynSetExpan: An iterative framework for joint entity set expansion and synonym discovery. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8292–8307, Online. Association for Computational Linguistics.

Jiaming Shen, Yunyi Zhang, Heng Ji, and Jiawei Han. 2021. Corpus-based open-domain event type induction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5427–5440, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 304–311, New York City, USA. Association for Computational Linguistics.

Étienne Simon, Vincent Guigue, and Benjamin Piwowarski. 2019. Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1378–1387, Florence, Italy. Association for Computational Linguistics.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2020. Revisiting unsupervised relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7498–7505, Online. Association for Computational Linguistics.

Ruidong Wu, Yuan Yao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. 2019. Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 219–228, Hong Kong, China. Association for Computational Linguistics.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 712–720, Jeju Island, Korea. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Dian Yu, Lifu Huang, and Heng Ji. 2017. Open relation extraction and grounding. In *Proc. the 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

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

Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. A relation-oriented clustering method for open relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9707–9718, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Experiment Details

We use an effective batch size of 32 (among $\{8, 16, 32, 64\}$) and train with an initial learning rate of $5e{-}5$ (among $\{1e{-}5, 3e{-}5, 5e{-}5, 1e{-}5\}$) using the AdamW optimizer and a linear schedule. The model is pretrained for 3 epochs for initialization and then further trained for 30 epochs on TACRED/ACE and 20 epochs on FewRel. For the hyperparameters in our model, we set the margin for the hinge loss $\alpha = 2$ following (Hsu et al., 2018). We show some additional tuning results in Table 11. The weight for the consistency loss $\beta = 0.2$ was tuned from $\{0.1, 0.2, 0.5\}$. We tuned our hyperparameters on TACRED based on accuracy and applied them to FewRel and ACE.

Our models are trained on a single Nvidia RTX A6000 GPU. A single run on TACRED takes 2 hours, a run on FewRel takes 2.5 hours and a run on ACE takes 40 minutes. Our model has 111M parameters (110M are from bert-base).

B Varying Cluster Number K

In Figure 4 and 5 we show how the model's performance changes with different specified number of unknown types K. Generally speaking, K will impact the granularity of the discovered types. On the ACE dataset, a slightly larger number of K will lead to improved performance. At $K = 35$, the model is able to separate `Business:End-Org` from `Business:Merge-Org` which were originally mixed at $K* = 23$. On TACRED, though, $K* = 10$ seems to be the optimal value, and a larger $K = 20$ would result in `per:cause_of_death` being split into subcategories of disease, homicide, accident and `per: charges` being split into subcategories of violent (e.g. murder) and non-violent (e.g. espionage).

α	Acc	B^3
0.5	0.9063	0.8841
1	0.8696	0.8493
2	0.9030	0.8806
5	0.9063	0.8842

Table 11: Tuning the hinge loss margin α on TACRED.

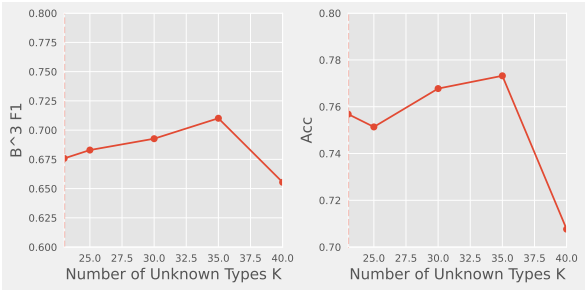

Figure 4: Performance of type discovery on ACE with varying cluster number K. The ground truth number of clusters $K = 23$.

C Human Evaluation Protocol

We evaluate the end-to-end results for event discovery from both the cluster level and instance level. For each cluster, we present the top 10 and bottom 10 instances and ask annotators if this cluster is meaningful and relevant to the corpus. For instance-level evaluation, we ask the annotator whether an instance belongs to a set of candidate instances or

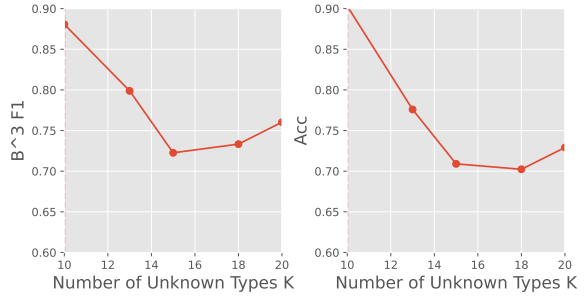

Figure 5: Performance of type discovery on TACRED with varying cluster number K. The ground truth number of clusters $K = 10$.

not. This set of candidate instances is either sampled from the same predicted cluster or randomly selected from other clusters with 50% probability.

D End-to-end Event Discovery Case Study

In Table 12 we show the results of our model along with ETypeClus under the end-to-end setting. The pipeline of ETypeClus converts predicate mentions into predicate-object (P-O) pairs, selects salient P-O pair then clusters such salient pairs. As a result, the output clusters do not cover infrequent triggers such as "swinging" and "siphoning" and the clusters themselves are often tied together by shared predicates or shared objects (`establish state`, `establish administration` and `endorse administration`). Our model, on the other hand, operates directly on predicate mentions, allowing us to identify events with infrequent triggers and events with named entity or pronoun objects as in "set up EasyJet" and "blame each other".

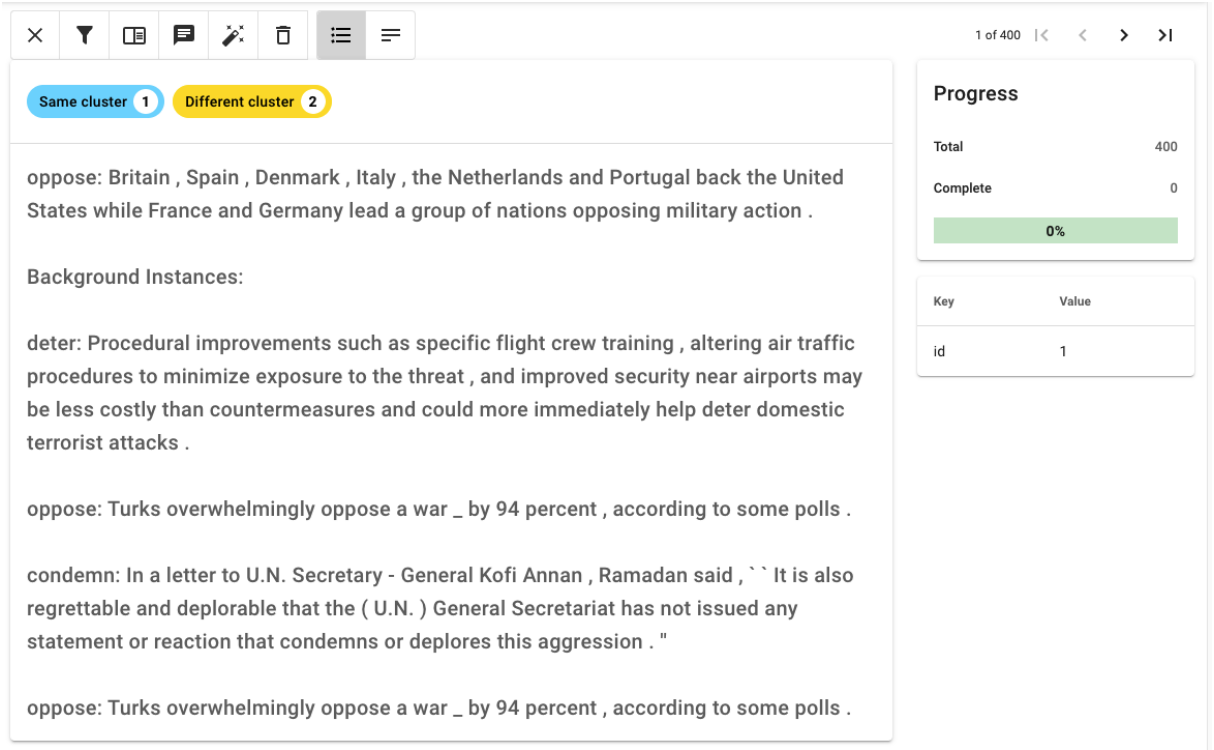

Figure 6: An example of the evaluation interface presented to annotators.

Event Type	ETypeClus **Predicate-Obj**	Ours **Predicate**	Ours **Mentions**
Transaction	return-1 piece, sell-3 cookie, sell-3 commercial, buy-0 pudding, sell-0 park, build-0 housing, sell-5 share	sell	The program allows Iraq to **sell** unlimited quantities of oil to buy food
		buying	They're basically **buying** future medical care throughout their lives
		swinging	Motorola and Texas Instruments both in the chips base **swinging** to profits
		siphoning	He had also been accused of **siphoning** millions of dollars from Project Coast to finance a lavish, globe-trotting lifestyle
Create	build-2 blog, establish-0 country, form-2 group, endorse-1 administration, incorporate-0 blog, establish-0 state, establish-0 administration	create	Major US insurance group AIG is in the final stage of talks ... in a deal to **create** Japan 's sixth largest life insurer
		produce	The electricity that Enron **produced** was so exorbitant that the government decided it was cheaper not to buy electricity
		set (up)	EasyCinema founder Stelios Haji - Ioannou , who **set up** easyJet in 1995
		pass	U.S. Ambassador John Negroponte was asked whether the United States would withdraw the resolution if it didn't have the votes to **pass** it
Oppose	maintain-1 innocence, plead-2 conspiracy, denounce-0 move, reject-0 change, rid-0 move, oppose-0 move, announce-2 creation, denounce-1 presence	rejecting	the flight attendants came in with a close vote **rejecting** these concessions
		opposed	78 of 100 people surveyed **opposed** the military action in Iraq
		blamed	A summit ... had been planned for Wednesday but was postponed, according to Israeli and Palestinian officials , who **blamed** each other for the delay.
		objected	Russia **objected** to World Bank rules that required monitoring of patients receiving medication

Table 12: Discovered type clusters in the end-to-end setting on ACE. The event type names were manually assigned based on the cluster content. The predicate mentions are in bold. The questionable assignments are underlined.