# Semi-supervised New Event Type Induction and Description via Contrastive Loss-Enforced Batch Attention

**Carl Edwards** and **Heng Ji**
University of Illinois Urbana-Champaign
{cne2, hengji}@illinois.edu

## Abstract

Most event extraction methods have traditionally relied on an annotated set of event types. However, creating event ontologies and annotating supervised training data are expensive and time-consuming. Previous work has proposed semi-supervised approaches which leverage seen (annotated) types to learn how to automatically discover new event types. State-of-the-art methods, both semi-supervised or fully unsupervised, use a form of reconstruction loss on specific tokens in a context. In contrast, we present a novel approach to semi-supervised new event type induction using a masked contrastive loss, which learns similarities between event mentions by enforcing an attention mechanism over the data minibatch. We further disentangle the discovered clusters by approximating the underlying manifolds in the data, which allows us to achieve an adjusted rand index score of 48.85%. Building on these clustering results, we extend our approach to two new tasks: predicting the type name of the discovered clusters and linking them to FrameNet frames.[1]

## 1 Introduction

Discovering new event types is an important step for adapting information extraction (IE) methods to unseen domains. Existing work (Ji and Grishman, 2008; McClosky et al., 2011; Li et al., 2013; Chen et al., 2015; Du and Cardie, 2020; Li et al., 2021a) traditionally uses a predefined list of event types and their respective annotations to learn an event extraction model. However, these annotations are both expensive and time-consuming to create. This problem is amplified when considering specialization-intensive domains such as scientific literature, which requires years of specialized experience to understand even a specific niche. For example, there are a wide range of otherwise obscure events in biomedical literature (Krallinger et al., 2017), and better IE techniques can empower life-changing breakthroughs in these domains. To adapt IE to these specialized domains, it is critical to discover new event types automatically.

There are two primary approaches in event type induction. The first is completely unsupervised induction. It includes recent neural techniques (Huang et al., 2016; Shen et al., 2021), as well as ad-hoc clustering techniques (Sekine, 2006; Chambers and Jurafsky, 2011) and probabilistic generative methods (Cheung et al., 2013; Chambers, 2013; Nguyen et al., 2015). The second approach, semi-supervised event type induction, was recently introduced by Huang and Ji (2020). It proposes leveraging annotations for existing types to learn to discover new types; this enables taking advantage of existing resources. In this work, we pursue the second approach.

Current state-of-the-art work in event type induction (Huang and Ji, 2020; Shen et al., 2021) uses reconstruction-based losses to find clusters of new types. Motivated by recent success in learning representations with contrastive loss (Chen et al., 2020a; Radford et al., 2021), we propose a novel alternative approach using batch attention and contrastive loss, which achieves state-of-the-art results. Essentially, we consider the attention weight between two event mentions as a learned similarity, and we ensure that the attention mechanism learns to align similar events using a semi-supervised contrastive loss. By doing this, we are able to leverage the large variety of semantic information in pretrained language models for clustering unseen types by using a trained attention head. This reveals our first key motivation: unlike (Huang and Ji, 2020), we are able to separate clustering from learning, allowing specific task-suited clustering algorithms to be selected. This allows us to test multiple clustering strategies after training once.

[1] The programs and resources will be publicly available at github.com/cnedwards/EventTypeBatchAttention for research purposes.

*Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3805–3827
May 2-6, 2023 ©2023 Association for Computational Linguistics

Additionally, this easily allows the use of hierarchical clustering, which may be beneficial to some downstream tasks.

Batch attention is an attention mechanism taken over a minibatch of samples rather than a sequence. Previous uses of batch attention have been limited. Primarily, it has been used for image classification (Cheng et al., 2021) and satellite imagery (Su et al., 2019). In this work, we apply batch attention to natural language instead, which we use for clustering, and we propose the novel idea of enforcing the attention mechanism using contrastive loss.

To enable our discovered event types to be used in larger IE systems, it is important to extract information regarding the clusters. Previous work has looked to describe clusters—for a given cluster, Huang et al. (2016) uses the nearest trigger to the cluster centroid as its name. However, this approach is nebulous and not easily measurable (because the same trigger can correspond to different event types and there is not a quantitative method to determine if the selected trigger defines the cluster well). This is our second key motivation: we instead introduce two new information retrieval-styled tasks for describing the cluster – type name prediction and FrameNet (Baker et al., 1998) frame linking. Type prediction predicts a name for each cluster and is a relatively easy task. FrameNet linking builds on this by linking event types to relevant frames, and is significantly more useful for downstream applications. Our attention-based approach is especially useful here, since it uses the attention mechanism to produce "clustered" features which can have auxiliary task-specific losses applied (prior work is not well-suited for this because the loss is applied to individual data points). This allows our method to build clusters which are more amenable to downstream tasks.

The major novel contributions of this paper are:

- We propose a novel framework for new event type induction which uses a novel masked contrastive loss to enforce an attention mechanism over data minibatches. This framework is also potentially applicable for semi-supervised clustering and classification problems in other settings where a pretrained model exists.

- We show that the base pretrained model selected for event type induction plays a key role in the types which are discovered, since even un-finetuned models rival Huang and Ji (2020).

- We use the "clustered" features produced by our model to extend new event type induction to two novel downstream tasks: type name prediction and FrameNet linking. We show our architecture design allows for auxiliary losses which improve performance on these tasks.

## 2 Task Descriptions

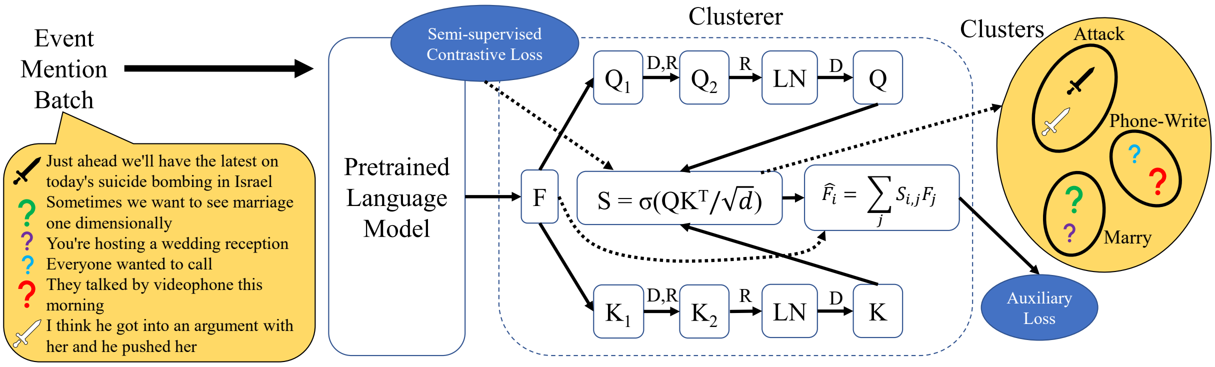
### 2.1 Semi-supervised Event Type Induction

We tackle the problem of semi-supervised event type induction, first described by Huang and Ji (2020). The task is defined as follows: Assume that $k$ event types from some dataset are known and that the types of the other events are unknown. Using the known types as example clusters, we seek to discover type clusters for the unknown types. Essentially, this is a semi-supervised clustering task on event mentions. In this work, we follow Huang et al. (2018) and set the 10 most common types in the ACE 2005 dataset (Walker et al., 2005) as known. Thus, given all ACE annotated event mentions, our goal is to automatically discover the other 23 unseen ACE types. This assumption is likely to carry over to real-world datasets, since existing 'seen' type definitions are most likely to cover the most common events. In many real-world cases, there may be a few long-tail event types present in the 'seen' types and one or two very common event types may be 'unseen'. Regardless, the distribution of seen and unseen types is likely to be fairly similar to the distribution of setting the most popular $k$ types as known.

### 2.2 Downstream Clustering Tasks

Beyond clustering, we also introduce two new downstream tasks on this problem: type prediction and FrameNet (Baker et al., 1998) linking. We structure both of these tasks as information retrieval problems for evaluation. Essentially, given a cluster, one should be able to predict its event type name and to what frame it should be linked. For each cluster, we calculate the most frequent type and consider it to be the ground truth for the cluster.

**Type Prediction**: The goal is to retrieve the "name" of the correct type for a cluster. Thus, we measure Hits@n and mean reciprical rank (MRR), where the corpus consists of the 23 new unseen type names.


Event Mention Batch
Semi-supervised Contrastive Loss
Clusterer
Clusters
Attack
Phone-Write
Marry
Pretrained Language Model
F
$Q_1$
D,R
$Q_2$
R
LN
D
Q
$S = \sigma(QK^T/\sqrt{d})$
$\hat{F}_i = \sum_j s_{i,j}F_j$
$K_1$
D,R
$K_2$
R
LN
D
K
Auxiliary Loss

Just ahead we'll have the latest on today's suicide bombing in Israel

Sometimes we want to see marriage one dimensionally

You're hosting a wedding reception

Everyone wanted to call

They talked by videophone this morning

I think he got into an argument with her and he pushed her

Figure 1: Architecture of the proposed approach. Best viewed in color. LN is layer normalization, R is ReLU, and D is dropout. $\sigma$ is softmax for the attention mechanism and sigmoid for the contrastive loss. $\hat{F}_i$ is the clustered features of mention $i$ in the batch. '**?**' are unseen event types.

In practice, we embed the names using our language model and use cosine similarity to the cluster centroid to rank them.

**FrameNet Linking**: FrameNet is the largest event ontology that is publicly available. However, there is not enough annotated training data to train supervised models directly on it. To alleviate this issue, we propose a task linking our newly discovered event types to FrameNet frames.

For the FrameNet linking, we consider a setup similar to name prediction, where we link clusters to the 1,221 frames in FrameNet 1.7 (Ruppenhofer et al., 2016). Instead of using the type names, we follow Huang et al. (2018) and manually map the ACE types to one or more frames to create a ground truth (see Appendix C for details). All child frames of the mapped frames are also considered valid targets. Given an ACE type, we can now link to a set of valid frames. In practice, we take the corpus of frame definitions and embed them using our language model. We then rank them using cosine similarity by comparing to the given cluster centroid. We consider the best rank of the valid frames to be the rank of a cluster. Thus, this task is also measured with Hits@n and MRR.

# 3 Methods

## 3.1 Overall Architecture

Overall, our method, shown in Figure 1, consists of a language model, such as BERT (Devlin et al., 2019), which produces contextualized representations, followed by a "clusterer". Unlike previous work which used specific token representations (Huang and Ji, 2020), we use mean pooling over the entire mention where an event occurs as our input. The language model produces an event representation, which is then input into the "clusterer" layer. The clusterer layer then produces "clustered" features using the attentions (see Section 3.3).

## 3.2 Back-translation for Data Augmentation

Contrastive loss has recently been applied for deep clustering (Li et al., 2021b; Zhong et al., 2020) and for representation learning (Chen et al., 2020a; Gao et al., 2021; Zhang et al., 2021a; Liu and Liu, 2021). However, this requires data augmentation to create positive example pairs. For text, some augmentations use back-translation (Cao and Wang, 2021; Zhang et al., 2021b). Taking inspiration from these clustering and representation learning techniques, we employ back-translation as data augmentation to create more positive pairs, improving the learning of attention weights between event mentions.

## 3.3 Batch Attention "Clusterer" Mechanism

To learn similarities between unseen event mentions, we propose learning an attention mechanism over the stochastic gradient descent minibatch. We enforce this attention mechanism using a masked contrastive loss (described in Section 3.4). This allows the attention mechanism's behavior to be learned from the seen classes.

We follow Vaswani et al. (2017) in implementing a scaled dot product attention, although over the batch instead. Since our "clusterer" needs to learn similarities for clustering and then be used for cluster features, we use nonlinear transformations for the query ($Q$) and key ($K$) vectors instead of the linear transformations in (Vaswani et al., 2017). This nonlinear transformation for $Q$ and $K$ is implemented as a two hidden layer neural network, which is shown in Figure 1.

Using this attention mechanism, we produce "clustered features", which are a convex combination of the different samples from the batch. This allows us to apply an auxiliary loss to the clustered features. We consider this as being analogous to learning on cluster centroids. Specific auxiliary losses can be applied for specific downstream tasks.

We note that this approach can also be interpreted as a type of feature smoothing, an inner product graph generator, and metric learning.

### 3.4 Masked Semi-supervised Contrastive Loss

Recent work, such as CLIP (Radford et al., 2021) and Text2Mol (Edwards et al., 2021), has found great success using contrastive losses between pairs of representations $Q$ and $K$, each $n \times d$ matrices where $n$ is the number of samples of $d$ dimensions. They obtain the loss $L$ by comparing the product of these matrices ($QK^T$) to a label matrix $Y \in \{0,1\}^{n \times n}$ (which in their case is $Y = I_n$), using cross entropy loss $CE$.

$$L(Q,K) = CE(QK^T, I_n) + CE(KQ^T, I_n)$$

We modify these existing contrastive losses to enforce our batch attention mechanism. We calculate the label matrix as follows (see Appendix B for an example). Given a pair of samples (event mentions) $x_i$ and $x_j$, we consider the pair to be a positive if they are from the same seen event type. We consider the pair to be negative if they are from different seen event types or if one is seen and one unseen. If both are unseen they are masked. In practice, the labels can be computed using one-hot vectors of the $c$ seen types (the unseen types are zero-hot vectors). These vectors are stacked into a $n \times c$ matrix $O$. The label matrix is computed

$$Y = OO^T \vee I_n$$

where $\vee$ is the elementwise logical-or operation. Following (Edwards et al., 2021), we use binary cross-entropy as the loss between the labels $Y$ and the scaled attention dot products $\frac{QK^T}{\sqrt{d}}$ from (Vaswani et al., 2017). This gives the following contrastive loss:

$$L_{ss}(Q,K) = CE(\frac{QK^T}{\sqrt{d}}, Y)$$

This loss, however, values negative samples much more than positive samples (due to the imbalance). Noticing that once vectors of a negative pair are orthogonal they don't need to be further separated, we introduce a margin $m$. Essentially, we mask out pairs whose dot product is "too negative" (in addition to unknown relations between unseen types). This is because the loss would rather optimize the already well-separated negatives instead of the relatively fewer positives. Let $p_{i,j} \in \{0,1\}$ be the label of a pair and $u_{i,j} \in \{0,1\}$ indicate that both $i$ and $j$ are unseen. Our mask, $M$, is calculated

$$M_{i,j} = \overline{\left[\overline{p_{i,j}} \wedge \left(\sigma(\frac{QK^T}{\sqrt{d}})_{i,j} < m\right)\right] \vee u_{i,j}}$$

where $\wedge$ is elementwise logical-and, $\sigma$ is the sigmoid function, and $\bar{z}$ denotes logical negation of $z$. This is similarly motivated to the margins used in knowledge graph embedding losses, such as TransE (Bordes et al., 2013). Thus, our loss is:

$$L_m(Q,K) = M \cdot L_{ss}(Q,K)$$

where in this case we treat $L_{ss}(Q,K)$ as an unreduced loss (so it is a matrix), and $\cdot$ is elementwise multiplication.

We apply this loss to the query ($Q$) and key ($K$) matrices in the clusterer's batch attention mechanism. We also include the augmented data ($Q'$ and $K'$), giving us a final loss:

$$L_c(Q,K,Q',K') = \sum_{\hat{Q},\hat{K} \in \{Q,Q'\} \times \{K,K'\}} L_m(\hat{Q},\hat{K})$$

### 3.5 Auxiliary Loss

For our downstream tasks, we employ a regression-based auxiliary loss. For each seen instance $x_i$, we maximize the cosine similarity between the clustered features $\hat{F}_i$ and the pretrained language model embedding $B_{t_i}$ of the ground truth type $t_i$ (e.g. the name 'attack'). Thus, we get the loss:

$$L_a(\hat{F}_i, B_{t_i}, t_i) = 1 - \cos(\hat{F}_i, B_{t_i}) \mathbb{1}_{\text{seen}}(t_i)$$

where $\mathbb{1}_{\text{seen}}(t_i)$ indicates whether $t_i$ is a seen type.

### 3.6 Stopping Criterion

For this task, it is not reasonable to use a validation set for stopping. This is because the loss depends only on seen types and their relationships to unseen types. Since the unseen classes are unlabeled and the losses between pairs of unseen are unknown, the model can overfit to the seen data, pushing together clusters of unseen types. To deal with this issue, we employ unsupervised clustering metrics to decide when to stop training. In particular, we use cosine distance-based silhouette scores to measure the quality of clustering. Further details are given in Appendix F.

### 3.7 Clustering

Any algorithm which can compute clusters from a precomputed distance function can be applied to the learned similarities between event mentions. Additionally, we find that the finetuning of the language model by our loss modifies its representations to better form clusters. Thus, this representation can be used in many clustering algorithms as well.

#### 3.7.1 Manifold Approximation

Inspired by recent work (Ros et al., 2021) which uses manifold approximation to interpret large language model-based sentence representations for information retrieval, we incorporate manifold approximation into our clustering approach. To do so, we follow the UMAP (McInnes et al., 2018) algorithm to create approximate weights based on estimating neighborhood densities within the data. We calculate these weights using cosine distance as an input, as it has traditionally been effective for language modeling (Manning et al., 2008; Reimers and Gurevych, 2019). UMAP attempts to estimate the density by comparing the distance to the k-nearest neighbors. This is used to calculate weights between each pair of data points. Details are given in Appendix H. Following this, we use agglomerative clustering on the UMAP weights as before.

In our approach, we want to better understand the global clustering landscape, so we use a high value of $k$. In practice, to avoid hyperparameter selection, we set $k$ equal to the size of the data.

## 4 Experimental Results

Generally, we used default hyperparameters. We split the learning rates into BERT and non-BERT parameters following (Edwards et al., 2021) with 2e-5 for BERT as in (Devlin et al., 2019) and 1e-4 for other parameters as in (Vaswani et al., 2017). For the margin parameter, we examined silhouette scores to select 0.5.

For back-translation, we used four languages, German, French, Spanish, and Chinese, and randomly sampled which language to use for each data point every epoch. We obtained back-translations using the MarianMT translation models (Junczys-Dowmunt et al., 2018). Ablations are shown in Section J.

For our main experiments, we only use the contrastive loss. We take the average of 5 runs to show that our method consistently outperforms (Huang and Ji, 2020). We also calculate clusters using an ensemble of the 5 runs which shows slightly increased performance, which is an expected result in deep neural networks (Allen-Zhu and Li, 2020).

Huang and Ji (2020) evaluate these clusters using Geometric NMI, Fowlkes Mallows (Fowlkes and Mallows, 1983), Completeness, Homogeneity, and V-Measure (Rosenberg and Hirschberg, 2007). We additionally consider adjusted Rand index (ARI) (Hubert and Arabie, 1985). In the downstream tasks, given a clustering we also report the average cluster purity and type representation. Given a cluster $i$ of size $n_i$ with most frequent type numbering $n_{f_i}$, purity $p_i = \frac{n_i}{n_{f_i}}$ (Manning et al., 2008). Note that this average cluster purity is slightly different than traditional purity; it weights small clusters more which is desirable in our case (like macro vs. micro F1 score). Type representation is the number of unique frequent subtypes, $n_t$, divided by total types, in this case 23.

### 4.1 Language Model

We select Sentence BERT (SBERT) (Reimers and Gurevych, 2019) as a language model because its pretraining tasks are better suited for clustering than BERT. This is shown in Table 1, since the clustering from SBERT embeddings can even outperform (Huang and Ji, 2020) without any semi-supervision. We use a small version of the model[2] from HuggingFace (Wolf et al., 2020), which allows us to use a larger minibatch size of 10. Using larger minibatch sizes is desirable for contrastive loss since the number of negatives scales quadratically with the size. The performance of mini SBERT is notable, as Huang and Ji (2020) used BERT-large, a considerably larger model.

### 4.2 Clustering Algorithms

For clustering, we consider two algorithms which work on precomputed metrics. First, we use agglomerative clustering with average linkage, as it tends to be less sensitive to outliers and noisy data (Han et al., 2011). Noise is present in the dataset, often in the form of transcripts (see Section 4.4).

We report results following existing clustering literature by using the true number of classes as the cluster number (Huang et al., 2020; Li et al., 2021b). In practice it is generally difficult to select the correct number of clusters to use. Due to this, using extra clusters is typically done by previous

[2] paraphrase-MiniLM-L12-v2

| Method | | Clusters | Geometric NMI | Fowlkes Mallows | Completeness | Homogeneity | V-Measure | ARI |
|---|---|---|---|---|---|---|---|---|
| One Cluster | | 1 | 0.00 | 25.58 | 100.00 | 0.00 | 0.00 | 0.00 |
| SS-VQ-VAE w/o VAE (Huang and Ji, 2020) | | 500 | 33.45 | 25.54 | 42.76 | 26.17 | 32.47 | - |
| SS-VQ-VAE (Huang and Ji, 2020) | | 500 | 40.88 | 31.46 | 53.57 | 31.19 | 39.43 | - |
| SS-VQ-VAE + SBERT [3] | | 33 | 19.08 | 19.45 | 25.80 | 15.13 | 19.08 | 7.54 |
| SBERT | Agglo | 23 | 50.71 | 34.35 | 57.05 | 45.07 | 50.36 | 24.02 |
| SBERT | Manifold | 23 | 48.75 | 36.02 | 51.32 | 46.30 | 48.68 | 30.21 |
| Attn-Cosine | Agglo | 23 | 46.40 | 34.60 | 49.82 | 43.24 | 46.27 | 26.69 |
| Attn-DotProduct | Agglo | 23 | 50.17 | 37.48 | 53.50 | 47.06 | 50.06 | 30.13 |
| Attn | Manifold | 23 | 54.83 | 42.77 | 55.00 | 54.67 | 54.82 | 38.74 |
| FT-SBERT | Manifold | 23 | 60.28 | 50.63 | 60.19 | 60.37 | 60.28 | 47.24 |
| Attn-DotProduct | Affinity | 49-68 | 56.87 | 35.64 | 49.58 | 65.26 | 56.33 | 30.02 |
| Attn-Cosine | Affinity | 50-69 | 56.54 | 33.00 | 48.72 | 65.62 | 55.91 | 27.04 |
| E-Attn-DotProduct | Agglo | 23 | 56.50 | 43.26 | 59.62 | 53.54 | 56.41 | 37.02 |
| E-Attn | Agglo | 23 | 59.00 | 46.19 | 58.36 | 59.66 | 59.00 | 42.56 |
| E-FT-SBERT | Manifold | 23 | **63.56** | **52.10** | **63.11** | 64.01 | **63.56** | **48.85** |
| E-Attn-DotProduct | Affinity | 63 | 60.00 | 38.41 | 51.32 | **70.15** | 59.28 | 31.78 |

Table 1: New event type induction results (%)[4]. The first subcolumn is the input for clustering and the second is the clustering algorithm used. SBERT indicates the (unfinetuned) SBERT representations were used rather than our learned attentions (Attn). FT-SBERT representations are finetuned by our method. E stands for ensemble. Values are the average of 5 runs. Agglo is agglomerative clustering, Affinity is (Frey and Dueck, 2007), and Manifold is as described in Section 3.7.1. For affinity, each run can produce a slightly different number of clusters.

work (Huang and Ji, 2020; Shen et al., 2021). However, this can inflate the NMI score (Nguyen et al., 2009) and benefit qualitative evaluation because of the unbalanced classes in the dataset. As an example, given only 23 clusters (the ground truth), a large class such as 'Injure' splits into multiple smaller clusters, which causes rare event types to be merged. Results show that 19 / 23 types are represented by a cluster in the 50 cluster case versus only 16 / 23 in the 23 cluster case. This makes results appear better for more clusters. Silhouette scores are higher for 23 clusters, however.

Unlike existing work (Huang and Ji, 2020), the number of clusters is unimportant for our learning process and can be selected afterwords, such as by selecting a high number as in (Huang and Ji, 2020; Shen et al., 2021) or automatically with affinity propagation (Frey and Dueck, 2007). Affinity propagation selects exemplars to automatically determine the number of clusters. Our approach is especially useful here, since affinity propagation does not complete when applied to default SBERT representations but does when using our contrastive loss-enforced attentions.

## 4.3 Results

We compare our results with Huang and Ji (2020), who first introduced this task, in Table 1. We find that just our choice of language model outperforms the baseline. Also, using dot products is more effective for our learned attention metric than cosine distance, since dot product without normalization, as in our attention mechanism, indicates confidence of clustering a pair of samples together (This is because the contrastive loss uses sigmoid).

### 4.3.1 Manifold Approximation

We find manifold approximation to be very effective in our experiments. Intuitively, we understand this manifold approximation as untangling the cluster manifolds from each other in the high-dimensional representation space. Interestingly, the results using the finetuned SBERT representations perform better than the results on the learned similarities. We find this to be quite interesting, especially because the representations change an average of 0.6 cosine distance from their starting points, as shown in Appendix A. Our method causes SBERT to inherently learn representations more amenable for clustering.

While manifold approximation works well for clustering here, we note that using UMAP for clustering is considered controversial.[5] While it works well in many cases, there are potential issues with artifacts or false tearing of clusters. We leave analysis of the interaction between high-dimensional semantic spaces obtained from language models and manifold approximation to future work.

[3] (Huang and Ji, 2020) did not release their code, and we were unable to reproduce their results. Nonetheless, we apply their method to the SBERT representations based on the description in their paper. We also use an equivalent number of clusters to the ground truth, unlike their paper, which enables comparability to our method.

[4] (Huang and Ji, 2020) appears to have used the former scikit-learn default of geometric NMI, which is why their v-score doesn't equal arithmetic NMI.

[5] https://umap-learn.readthedocs.io/en/latest/clustering.html

① **Injure:** If those weren't gunshot wounds to cause the broken bones, do they know what caused the fractures

- **Injure:** More than 40 were injured
- **Injure:** There was no information on the identity of the injured person
- **Injure:** Sergeant Chuck Hagel was seriously wounded twice in Vietnam

② **Charge-Indict:** 56-year-old forry drake has been charged with interstate transport of a minor

- **Charge-Indict:** Ocalan, being tried in absentia, was indicted for entering the country illegally, a
- **Convict and Charge-Indict:** convicted oklahoma city bombing conspirator terry nichols will stand trial again on state murder charges
- **Appeal:** in the african nation of nigeria, an islamic court delayed the appeal of a woman condemned to death by stoning

③ **Marry:** My wife and I were guests at a wedding on the Carnival Legend on New Years Eve 2003

- **Marry and Divorce:** Giuliani, 58, proposed to Nathan, a former nurse, during a November business trip to Paris - five months after he finalized his divorce from Donna Hanover after 20 years of marriage
- **Merge-org:** So Oracle and Peoplesoft , who spent the last 18 months insulting one another in every imaginable way, are finally tying the knot

④ **Declare-Bankruptcy:** You need to speak to a bankruptcy attorney pronto; this is a bankruptcy matter, not a tax matter

- **Declare-Bankruptcy:** despite operating under bankruptcy laws, united posted the best on time performance
- **Declare-Bankruptcy:** That means that he received the shares while he was still in bankruptcy, which means that the shares were potentially assets that the trustee could use to pay off creditors

⑤ **Start-Org:** Kiichiro Toyoda founded the automaker in 1937, transforming the loom manufacturer started by his father into an automaker

- **Merge-Org:** I believe any neutral management consultant worth his or her salt would recommend a merger of the two organizations
- **End-Org:** It's a dying organization, and this will be just the jolt it needs for another couple decades of somnambulant staggering before being ultimately replaced by far more efficient companies

⑥ **Marry:** Either its bad or good

- **End-Org:** i felt t7ire was something else too, much history behind silver cross to end is now
- **Trial-Hearing:** Yeah, we're a pretty small town, so our newspaper covers it a lot
- **Trial-Hearing:** Yeah, because I was really -- I wasn't really following it that much because I was
- **Start-Position:** then when they're ready to breed they go to the wb

Figure 2: Cluster Examples: Injure, Charge-Indict, Marry, Bankruptcy, Start-Org, and Bad Data, respectively.

| Cluster Strength | Clusters |
|---|---|
| Very Strong<br>(> 80% Purity) | Injure, Sue, Phone-Write,<br>Declare-Bankruptcy, Demonstrate, Trial-Hearing |
| Strong<br>(60-80% Purity) | Be-Born, Start-Position,<br>Charge-Indict, Marry |
| Ok<br>(40-60% Purity) | Release-Parole, Appeal, Injure |
| Mixed<br>(20-40% Purity) | Convict, Fine, Trial-Hearing,<br>Start-Org, Start-Position, Charge-Indict |
| Small Clusters<br>(< 2 samples) | Trial-Hearing, Nominate,<br>Start-Position, Phone-Write |

Table 2: Clusters sorted into purity classes.

### 4.4 Qualitative Cluster Analysis

We analyze the clusters produced by our best result, the ensemble. We classify the clusters according to purity in Table 2. We show examples from numbered clusters in Figure 2. Certain types of clusters, such as Injure ① and Demonstrate ④, form very strong clusters. We believe this is likely related to their size and lack of overlap with other types. There are two common sources of error: the first is semantic overlap. Start-org, merge-org, and end-org tend to overlap ⑤. Marry and divorce also slightly overlap ③—in the 23 cluster case they merge into one cluster, but in the 50 cluster case they are separate. Most types of courtroom related events—Charge-Indict, Trial-Hearing, Convict, Release-Parole, Appeal, Execute, Acquit, Extradite—have some degree of overlap ②. Second, the other main source of errors is "duplicates". This occurs in our method because two or more events can occur in the same event mention ②, ③. Since our method does not account for triggers, it cannot distinguish between duplicate mentions with multiple triggers. Future work can address this issue by combining our method with an existing trigger-based method such as (Huang and Ji, 2020). We also find that our method clusters "junk" data together ⑥, which are usually from transcripts. Errors occasionally occur from metaphorical language, such as when companies are "married" ③. We show more detailed examples of these observations in Appendix E.

### 4.5 Downstream Tasks

For the downstream tasks, we use different clusterings and try to discover information about the clusters. As a baseline, we compare against default (not finetuned) SBERT clustering (Base-23) and ground truth (perfect) clusters. We compare these to our ensemble clustering. For type prediction, we use SBERT embeddings to compute cluster centroids and compare to the SBERT representation of the type name (e.g. 'injure'). For FrameNet linking, we use the frame definition instead of the name (e.g. "The words [...] describe situations in which an Agent or a Cause injures a Victim [...]"). We also use an auxiliary loss, $L_a$, which we apply to a 1-layer neural network on the clustered features $\hat{F}$. This extra layer is employed to allow multiple auxilliary losses: we leave those experiments for future work. We compare using these finetuned representations in addition to default SBERT. Results are shown in Tables 3 and 4.

We find that our ensemble clustering outperforms the default SBERT clustering, and that we are able to recover the event type 60% of the time. For the ground truth clusters, our finetuning with

| Clustering | Representation | Mean Rank | Hits@1 | Hits@3 | Hits@5 | Hits@10 | Hits@15 | MRR | Average Purity | Type Representation |
|---|---|---|---|---|---|---|---|---|---|---|
| Base-23 | Untuned | 5.17 | 34.8% | 47.8% | 60.9% | 82.6% | 100% | 0.477 | 25% | 47.8% |
| FT-Base-23 | Finetuned | 4.43 | 56.5% | 65.2% | 78.2% | 82.6% | 91.3% | 0.660 | 58.9% | 65.2% |
| Ensemble-23 | Untuned | 3.65 | 60.9% | 69.6% | 69.6% | 95.7% | 100% | 0.679 | 68.6% | 69.6% |
| Ensemble-23 | Finetuned | 5.13 | 56.5% | 65.2% | 69.6% | 87.0% | 87.0% | 0.650 | 68.6% | 69.6% |
| Ensemble-50 | Untuned | 4.40 | 56.0% | 60.0% | 68.0% | 90.0% | 96.0% | 0.630 | 69.3% | 82.6% |
| Perfect-23 | Untuned | 2.30 | 69.6% | 73.9% | 82.6% | 95.7% | 100% | 0.758 | 100% | 100% |
| Perfect-23 | Finetuned | 2.83 | 73.9% | 82.6% | 91.3% | 91.3% | 95.7% | 0.800 | 100% | 100% |

Table 3: Results for cluster to name prediction task with different representations and clusterings. Finetuned indicates finetuned SBERT representations from our contrastive + auxiliary loss are used (otherwise default SBERT). "-x" at the end is the number of clusters. Perfect indicates the ground truth clustering, Base/FT-Base is SBERT clusterings (default or finetuned with our auxiliary loss), and Ensemble is the clustering of our ensemble result. Type representation shows what percent of unseen types represent the majority of a cluster.

| Clustering | Representation | Mean Rank | Hits@1 | Hits@5 | Hits@10 | Hits@50 | Hits@100 | MRR | Average Purity | Type Representation |
|---|---|---|---|---|---|---|---|---|---|---|
| Base-23 | Untuned | 95.9 | 4.3% | 21.7% | 26.1% | 30.4% | 34.8% | 0.128 | 25% | 47.8% |
| FT-Base-23 | Finetuned | 156.9 | 30.4% | 30.4% | 34.8% | 43.5% | 47.8% | 0.336 | 57.4% | 65.2% |
| Ensemble-23 | Untuned | 72.7 | 17.4% | 30.4% | 39.1% | 47.8% | 65.2% | 0.264 | 68.6% | 69.6% |
| Ensemble-23 | Finetuned | 115.7 | 21.7% | 34.8% | 34.8% | 43.5% | 52.2% | 0.308 | 68.6% | 69.6% |
| Perfect-23 | Untuned | 15.9 | 26.1% | 39.1% | 52.2% | 65.2% | 73.9% | 0.374 | 100% | 100% |
| Perfect-23 | Finetuned | 42.7 | 47.8% | 56.5% | 60.9% | 69.6% | 69.6% | 0.539 | 100% | 100% |

Table 4: Results for cluster to frame linking task. See Table 3 for notation.

an auxiliary loss improves MRR and Hits@1 over the default SBERT representations. Frame linking is much more difficult, since there are 1,221 frames, but we are able to recover the correct frame for 30% of clusters, while default SBERT only achieves 4%. Notably, the auxiliary loss clustering (FT-Base-23) even outperforms our ensemble clustering, demonstrating the flexibility of our model architecture. Using perfect clustering, our finetuned model achieves nearly 50% Hits@1, doubling the performance of the default SBERT model. The finetuning loss allows the model to train on the combined cluster features, which approximates the centroid of a cluster. This promotes the potential cluster to be shaped so that its centroid is better suited for the downstream tasks–being most similar to the correct type name or frame representation.

## 5 Related Work

Although event extraction has long been studied (Grishman, 1997; Ji and Grishman, 2008; McClosky et al., 2011; Li et al., 2013; Chen et al., 2015; Du and Cardie, 2020; Li et al., 2021a), recent focus has turned towards discovering events without annotations. It includes recent neural techniques (Huang et al., 2016; Liu et al., 2019; Shen et al., 2021), as well as ad-hoc clustering techniques (Sekine, 2006; Chambers and Jurafsky, 2011; Yuan et al., 2018) and probabalistic generative methods (Cheung et al., 2013; Chambers, 2013; Nguyen et al., 2015). Semi-supervised event type induction was recently introduced by Huang and Ji (2020). Zero-shot event extraction frameworks, such as (Huang et al., 2018), can be used to perform event extraction on the newly discovered types. Recent work by Gao et al. (2022) uses a weakly-supervised contrastive learning-based clustering approach for event representation learning.

Several new unsupervised deep clustering approaches use contrastive loss for clustering images (Li et al., 2021b; Zhong et al., 2020) and text (Zhang et al., 2021b). These methods require data augmentation to create positive example pairs. Contrastive loss has also been applied to learn representations. SimCLR (Chen et al., 2020a,b) uses image augmentations for unsupervised representation learning. Follow-up work has applied this loss to natural language (Gao et al., 2021; Zhang et al., 2021a; Liu and Liu, 2021), with some augmentations being back-translated text (Cao and Wang, 2021). Gunel et al. (2021) use fully supervised contrastive loss to finetune language models.

Batch attention has been investigated a little in the literature, such as for satellite imagery prediction (Su et al., 2019) or image classification (Cheng et al., 2021); however, it has not been used to learn clustered features. Seidenschwarz et al. (2021) recently proposed a related idea for a message-passing network weighted by attention for clustering images. We instead directly use (contrastive loss-enforced) attention weights for clustering.

## 6 Conclusion and Future Work

In this work, we present an exciting new approach for event type induction, where we use contrastive loss to control the learning of a batch attention mechanism for both finding and learning about new cluster types. We also consider manifold approximation for clustering, and we introduce two new downstream tasks: type name prediction and FrameNet linking. This new approach opens several interesting problems for future work. First, this method can potentially be incorporated with reconstruction loss-based approaches, which might improve results or obviate the early stopping criterion. Alternatively, the stopping criterion can be integrated into a loss function for better stopping control. Notably, this would enable a two-step process of learning clusters and then performing knowledge distillation using those clusters (or an ensemble) while learning other desired losses. Future work can investigate the interaction of manifold approximation with large language models and integrate it directly into the clusterer subnetwork. It may be possible to use a heirarchy of event types for knowledge transfer from data-rich types to long-tail types. Finally, FrameNet linking can be extended to Wikidata Q-Node linking, which contains millions of nodes. Our approach may also be applicable in other modalities with strong pretrained models.

## 7 Limitations

In this work, we present a contrastive loss-based batch attention method for new event type induction. Like most modern event type induction methods, this relies on strong existing representations (which are typically pretrained in a self-supervised manner). Our additional testing on the much larger MAVEN dataset (Appendix K) indicates that selecting an appropriate representation is an important factor. In that case, SBERT representations are inappropriate for the event mentions in the dataset, but trigger representations are effective, and our method further improves those results. Future research on identifying pretraining methods for initializing representations will be useful for this method and task (along with many other models such as retrieval-based large language models (Guu et al., 2020; Borgeaud et al., 2021).

Requiring early stopping is a limitation to our newly proposed event type induction method which does not occur in existing reconstruction-based methods (those methods instead require a predetermined cluster number along with other limitations). While an early stopping method is required, our method is effective, and we believe that our new approach is still valuable because it offers a different approach for new event type induction. Additionally, future work should be able to build upon our work to remove early stopping. Doing so may also help obviate the dependence on the initial representation. In Section 6 we detail several possibilities for future work to further improve upon this work.

## Acknowledgements

We would like to thank the anonymous reviewers for helping us to improve this paper. This research is based upon work supported in part by U.S. DARPA KAIROS Program No. FA8750-19-2-1004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

## References

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *ArXiv preprint*, abs/2012.09816.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2021. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*.

Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization. *ArXiv preprint*, abs/2109.09209.

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. *ArXiv preprint*, abs/2003.04297.

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 (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Qishang Cheng, Hongliang Li, Qingbo Wu, and King Ngi Ngan. 2021. Baˆ 2m: A batch aware attention module for image classification. *ArXiv preprint*, abs/2103.15099.

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, 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.

Carl Edwards, ChengXiang Zhai, and Heng Ji. 2021. Text2Mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 595–607, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edward B Fowlkes and Colin L Mallows. 1983. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Jun Gao, Wei Wang, Changlong Yu, Huan Zhao, Wilfred Ng, and Ruifeng Xu. 2022. Improving event representation via simultaneous weakly supervised contrastive learning and clustering. *arXiv preprint arXiv:2203.07633*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv preprint*, abs/2104.08821.

Ralph Grishman. 1997. Information extraction: Techniques and challenges. In *International summer school on information extraction*, pages 10–27. Springer.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques*. Elsevier.

Jiabo Huang, Shaogang Gong, and Xiatian Zhu. 2020. Deep semantic clustering by partition confidence maximisation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8846–8855. IEEE.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany. Association for Computational Linguistics.

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.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

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

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.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Martin Krallinger, Martin Pérez-Pérez, Gael Pérez-Rodríguez, Aitor Blanco-Míguez, Florentino Fdez-Riverola, Salvador Capella-Gutierrez, Anália Lourenço, and Alfonso Valencia. 2017. The biocreative v. 5 evaluation workshop: tasks, organization, sessions and topics.

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 (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021a. 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.

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. 2021b. Contrastive clustering. In *2021 AAAI Conference on Artificial Intelligence (AAAI)*.

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.

Yixin Liu and Pengfei Liu. 2021. SimCLS: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 1065–1072, Online. Association for Computational Linguistics.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to information retrieval.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1635, Portland, Oregon, USA. Association for Computational Linguistics.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv preprint*, abs/1802.03426.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International*

*Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197, Beijing, China. Association for Computational Linguistics.

Xuan Vinh Nguyen, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 1073–1080. ACM.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kevin Ros, Carl Edwards, Heng Ji, and ChengXiang Zhai. 2021. Team skeletor at touché 2021: Argument retrieval and visualization for controversial questions. In *CEUR Workshop Proceedings*, volume 2936, pages 2441–2454. CEUR-WS.

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.

J Ruppenhofer, M Ellsworth, MRL Petruck, CR Johnson, CF Baker, and J Scheffczyk. 2016. Framenet ii: Extended theory and practice (revised november 1, 2016).

Jenny Denise Seidenschwarz, Ismail Elezi, and Laura Leal-Taixé. 2021. Learning intra-batch connections for deep metric learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 9410–9421. PMLR.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 731–738, Sydney, Australia. 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.

Yanzhou Su, Yongjian Wu, Min Wang, Feng Wang, and Jian Cheng. 2019. Semantic segmentation of high resolution remote sensing image based on batch-attention mechanism. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 3856–3859. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus. Technical Report LDC2006T06, Linguistic Data Consortium, Philadelphia.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A massive general domain event detection dataset. In *Proceedings of EMNLP 2020*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-schema event profiling for massive news corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 587–596. ACM.

Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021a. Pairwise supervised contrastive learning of sentence representations. *ArXiv preprint*, abs/2109.05424.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021b. Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430, Online. Association for Computational Linguistics.

Huasong Zhong, Chong Chen, Zhongming Jin, and Xian-Sheng Hua. 2020. Deep robust clustering by contrastive learning. *ArXiv preprint*, abs/2008.03030.

# A How much do SBERT representations change?

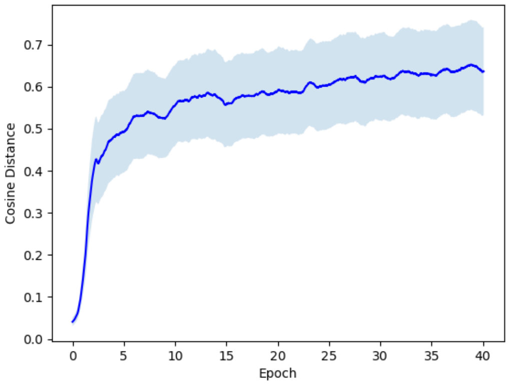


Figure 3: Change in SBERT representations from original representation of inputs. This shows that the representations change significantly from their starting point during finetuning. Shaded area is one standard deviation.

# B Label Matrix Example

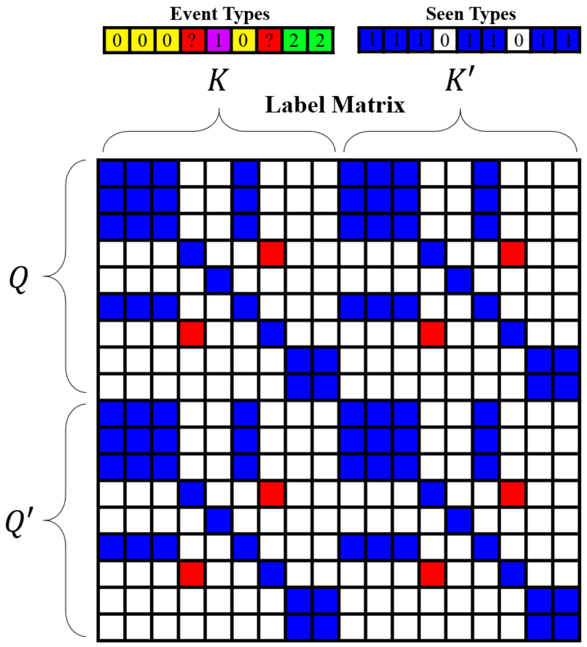


Figure 4: Best viewed in color. Visualization of the label matrix $Y$ used in the loss. Blue is positive, white is negative, and red is masked. Note that the mask for the negatives less than the margin is not shown. The event types and corresponding "seen" boolean vector are also shown, and are used to construct the label matrix. $Q$ and $K$ are corresponding queries and keys to the labels, while $Q'$ and $K'$ are augmented data.

# C Manual ACE05 to FrameNet Linking

| ACE Type | Frame |
|---|---|
| Appeal | Appeal |
| Be-Born | Birth_scenario |
| Charge-Indict | Notification_of_charges |
| Convict | Verdict |
| Declare-Bankruptcy | Wealthiness |
| Demonstrate | Protest |
| Divorce | Personal_relationship |
| End-Org | Organization ∣ Process_end |
| Extradite | Extradition |
| Fine | Fining |
| Injure | Cause_harm ∣ Experience_bodily_harm |
| Marry | Forming_relationships |
| Nominate | Appointing |
| Phone-Write | Contacting |
| Release-Parole | Releasing_from_custody |
| Start-Org | Organization ∣ Process_start |
| Start-Position | Being_employed ∣ Process_start |
| Sue | Judgment_communication |
| Trial-Hearing | Trial |
| Pardon | Pardon |
| Merge-Org | Organization ∣ Amalgamation |
| Acquit | Verdict |
| Execute | Execution |
| Attack | Attack |
| Transport | Transportation_status |
| Die | Death |
| Meet | Make_acquaintance ∣ Meet_with ∣ Come_together |
| Arrest-Jail | Arrest ∣ Prison ∣ Imprisonment ∣ Being_incarcerated |
| Sentence | Sentencing |
| Transfer-Money | Commerce_money-transfer |
| Elect | Change_of_leadership ∣ Choosing |
| Transfer-Ownership | Commerce_goods-transfer |
| End-Position | Being_employed ∣ Process_end |

Table 5: Mapping from ACE types to FrameNet frames. Some ACE types required multiple frames to be correctly mapped, which is indicated by " ∣ ".

# D Visualization

We visualize unseen event mentions using UMAP (McInnes et al., 2018) given a precomputed distance matrix of the cosine distance between $Q$ and $K$. Following (Huang and Ji, 2020), we show the results on six unseen types in Figure 6. Sentence and convict overlap significantly, which makes intuitive sense as they are semantically very similar. Unlike (Huang and Ji, 2020), trial-hearing forms its own cluster.

[6] Note that there is a mistake in (Huang and Ji, 2020), since "sentence" is a seen type in (Huang et al., 2018)

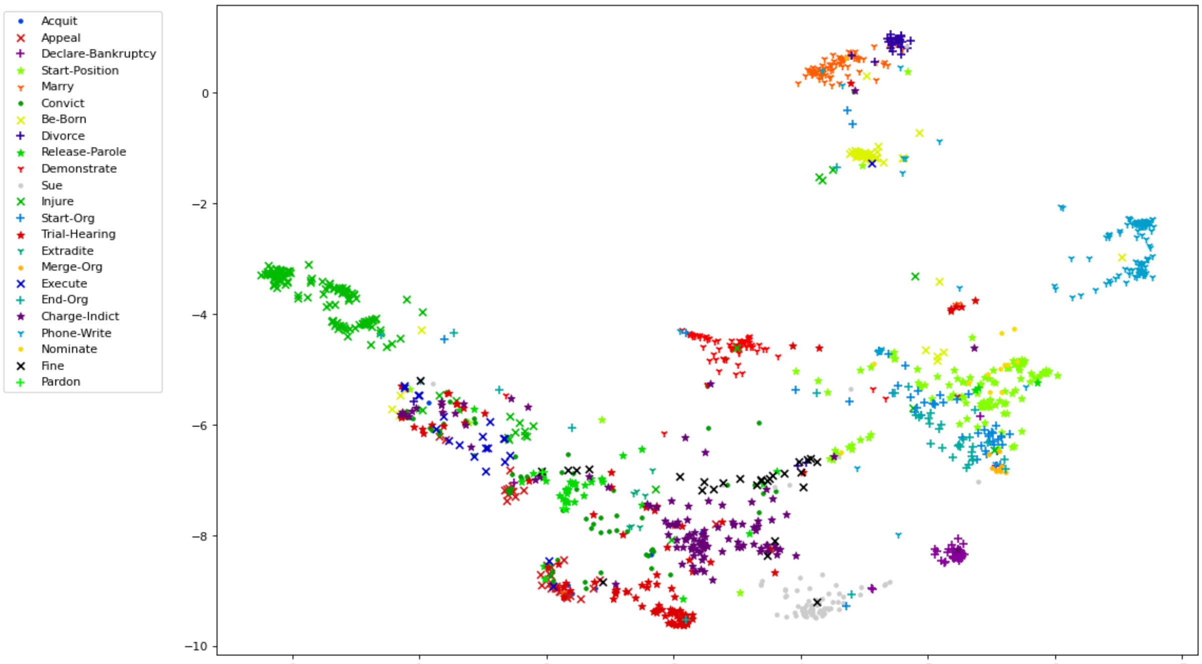


Figure 5: Visualization of all unseen types as seen by manifold approximation. Note that dimensionality reduction to 2D renders it difficult to understand with this high number of clusters, but the overall semantics of the space are interesting.

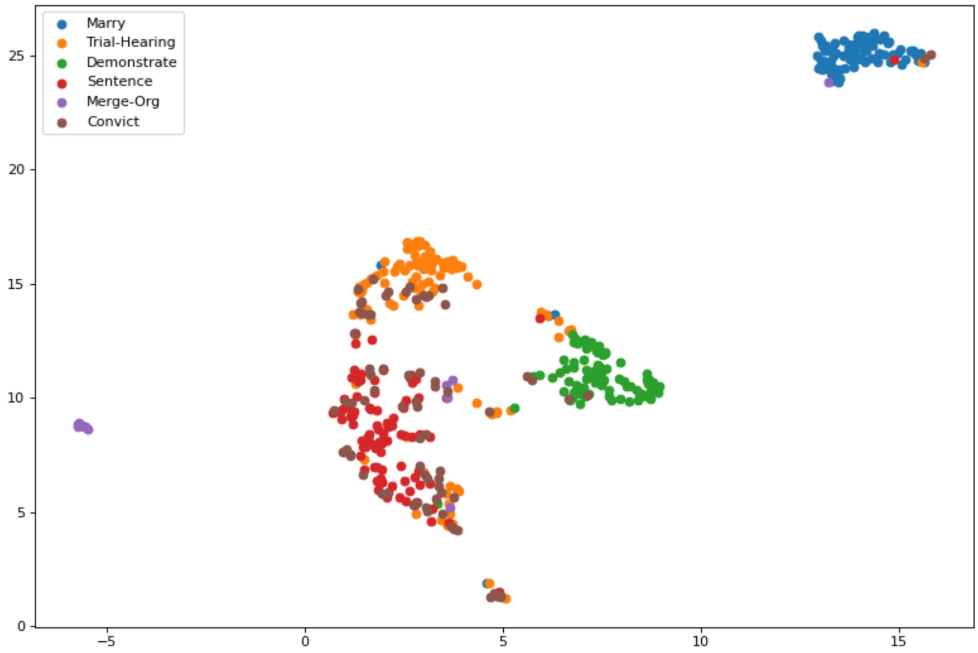


Figure 6: Visualization following (Huang and Ji, 2020) for one of the runs.[6] Note that our clusters have much less errors.

## E Cluster Examples

We show extensive examples of our noted observations in Tables 6 and 7. Namely, start-org, merge-org, and end-org tend to overlap. Marry and divorce slightly overlap in the 23 cluster case. Most types of courtroom related events—Charge-Indict, Trial-Hearing, Convict, Release-Parole, Appeal, Execute, Acquit, Extradite—have some degree of overlap. There are “duplicates” when two or more events can occur in the same event mention. We also note the cluster of “junk” data, where the label isn’t obvious from the event mention.

| Cluster Type | Purity | Cluster Member Example Types and Inputs |
|---|---|---|
| Injure | 98.3% | • **Injure:** According to other reports reaching here, five Syrian bus passengers were killed and 10 others were injured on Sunday morning when a US missile hit the bus they were traveling in near the Iraqi border<br>• **Injure:** More than 40 were injured<br>• **Injure:** There was no information on the identity of the injured person |
| Declare-Bankruptcy | 95.0% | • **Declare-Bankruptcy:** You need to speak to a bankruptcy attorney pronto; this is a bankruptcy matter, not a tax matter<br>• **Declare-Bankruptcy:** despite operating under bankruptcy laws, united posted the best on time performance<br>• **Declare-Bankruptcy:** That means that he received the shares while he was still in bankruptcy, which means that the shares were potentially assets that the trustee could use to pay off creditors |
| Demonstrate | 95.0% | • **Demonstrate:** The protest follows a string of others involving tens of thousands of peace activists across Japan since January<br>• **Demonstrate:** No, I don't demonstrate against anybody during a war<br>• **Demonstrate:** Several thousand demonstrators also gathered outside the White House in Washington, accompanied by a major security presence |
| Charge-Indict | 64.4% | • **Charge-Indict:** 56-year-old forry drake has been charged with interstate transport of a minor<br>• **Charge-Indict:** Ocalan, being tried in absentia, was indicted for entering the country illegally, a misdemeanor<br>• **Convict and Charge-Indict:** convicted oklahoma city bombing conspirator terry nichols will stand trial again on state murder charges<br>• **Appeal:** in the african nation of nigeria, an islamic court delayed the appeal of a woman condemned to death by stoning |
| Start-Position | 64.4% | • **Start-Position:** Many Iraqis boycotted the meeting in opposition to U.S. plans to install Garner atop an interim administration<br>• **Start-Position:** The meeting was Shalom's first encounter with an Arab counterpart since he took office as Israel's foreign minister on February 27<br>• **Start-Org:** Meeting in the biblical birthplace of the prophet Abraham, delegates from Iraq's many factions discussed the role of religion in the future government and ways to rebuild the country |

Table 6: Examples of discovered clusters. Charge-Indict shows an example of a duplicate—an input with multiple event types. It also shows how courtroom related events can overlap. For Start-Position, there are some errors related to the Middle East, which occurs frequently in the Start-Position mentions.

| Cluster Type | Purity | Cluster Member Example Types and Inputs |
|---|---|---|
| Marry | 70.2% | • **Marry:** My wife and I were guests at a wedding on the Carnival Legend on New Years Eve 2003<br>• **Marry and Divorce:** Giuliani, 58, proposed to Nathan, a former nurse, during a November business trip to Paris - five months after he finalized his divorce from Donna Hanover after 20 years of marriage<br>• **Phone-Write:** All the guests were folks who had met the bride and groom (an attractive young couple who were sailing alone) virtually on cruisecritic |
| Start-Org | 34.7% | • **Start-Org:** Kiichiro Toyoda founded the automaker in 1937, transforming the loom manufacturer started by his father into an automaker<br>• **Merge-Org:** I believe any neutral management consultant worth his or her salt would recommend a merger of the two organizations<br>• **End-Org:** It's a dying organization, and this will be just the jolt it needs for another couple decades of somnambulant staggering before being ultimately replaced by far more efficient companies |
| Bad Data | - | • **Marry:** Either its bad or good<br>• **End-Org:** i felt t7ire was something else too, much history behind silver cross to end is now<br>• **Trial-Hearing:** Yeah, we're a pretty small town, so our newspaper covers it a lot |
| Phone-Write | 86.2% | • **Phone-Write:** Let's see, my first call I got was from Russia<br>• **Phone-Write:** I'm chewing gum and talking on the phone while writing this note<br>• **Phone-Write:** He wants to call his mom in Houston |
| Sue | 92.5% | • **Sue:** Buyers and sellers also would have to agree not to pursue further cases in foreign courts<br>• **Sue:** The cost of class actions is factored into the cost of everything you buy<br>• **Sue:** The average number of suits against a neurosurgeon is five in South Florida |

Table 7: More examples of discovered clusters. Start-Org shows the semantic overlap between the organization-related clusters. Bad Data shows a cluster which mostly contains unclear input.

# F Early Stopping

For this task, it is not reasonable to use a validation set for stopping. This is because the loss depends only on seen types and their relationships to unseen types. Since the unseen classes are unlabeled and the losses between pairs of unseen are unknown, the model can overfit to the seen data, pushing together clusters of unseen types. We partially address this issue by implementing a margin on negative values, which prevents the model from forcing together unseen clusters as strongly to separate them from seen type events. Because of this, our method requires strong pretrained representations to build on, which have luckily become common in recent years. To deal with this issue, we employ unsupervised clustering metrics to decide when to stop training. In particular, we use cosine distance-based silhouette scores to measure the quality of clustering. This increases the required compute up to 2x (in practice roughly 1.5x because back-propagation isn't required), but training is already relatively quick, with 10 or less epochs being sufficient. We note that this approach can have some variance. To address this issue, we employ a sliding window running average approximation to create a smooth curve of the initial increase and then decrease of the silhouette score. We consider a hybrid approach—we select the window with the highest silhouette score, and then we select the epoch with the highest silhouette score in that window as our stopping point, as shown in Figure 7.

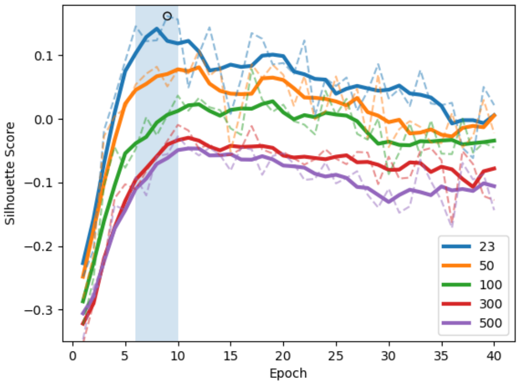


Figure 7: Bold lines are sliding window averages of size 5 over silhouette scores. Dotted lines are unsmoothed scores. Legend shows number of clusters. Note that the silhouette scores initially increase and then decay as overfitting occurs, resulting in the need for early stopping. Here, for 23 clusters, epoch 8 has the highest average score. The blue region shows the window around it, and epoch 9 (the black dot) is selected for stopping.

# G Evaluation Metrics

For the information retrieval metrics, given a list of rankings $R$,

$$MeanRank = \frac{1}{n}\sum_{i=1}^{n} R_i$$

$$MRR = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{R_i}$$

$$Hits@m = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{R_i \leq m}$$

## G.1 Clustering Evaluation Metrics

Assume there are two clusterings: a set of (ground truth) classes $C$ and a set of (predicted) clusters $K$. Each have $N$ samples. Denote $TP$ as true positives, the number of data point pairs that are in the same cluster in $C$ and $K$. $FP$ is the false positives, the number of data point pairs that belong in the same cluster in $C$ but are not in $K$. $FN$ is false negatives, the number of data point pairs that are in the same cluster $K$ but not in the same ground truth cluster in $C$. $TN$ is the number of data point pairs that are in different clusters in both $C$ and $K$. scikit-learn (Pedregosa et al., 2011) is used to compute scores.

- **Geometric NMI** is the normalized mutual information between two cluster assignments. It is defined:

$$NMI = \frac{I(C,K)}{mean(H(C),H(K))}$$

where I is the mutual information and H is entropy. In this case, $mean$ is the geometric mean.

$$mean(x_1,...,x_n) = \left(\prod_{i=1}^{n} x_n\right)^{\frac{1}{n}}$$

We note that arithmetic NMI using the arithmetic mean is often reported, but that it is equivalent to V-Measure.

- **Fowlkes Mallows** (Fowlkes and Mallows, 1983) is used to evaluate the similarity between a clustering and the ground truth. It is the geometric mean of pairwise precision and recall.

$$FM = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}}$$

- **Completeness** (Rosenberg and Hirschberg, 2007) Completeness measures whether all of the data points assigned to a single class are assigned to a single cluster. It is defined:

$$c = \begin{cases} 1 & \text{if } H(K, C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases}$$

- **Homogeneity** (Rosenberg and Hirschberg, 2007) measures whether data points in a cluster are all assigned the the same class. It is symmetric to completeness:

$$h = \begin{cases} 1 & \text{if } H(C, K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases}$$

- **V-Measure** (Rosenberg and Hirschberg, 2007) (standing for validity) is the harmonic mean between homogeneity and completeness:

$$v = \frac{(1 + \beta)hc}{\beta h + c}$$

In practice, $\beta = 1$ is used to weight $h$ and $c$ equally.

- **Adjusted Rand Index** (Hubert and Arabie, 1985) is a version of the Rand index, a measure of cluster similarity, which is adjusted for chance.

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max RI - \mathbb{E}[RI]}$$

where the Rand index, $RI$, is

$$RI = \frac{TP + TN}{\binom{n}{2}}$$

and $\mathbb{E}[RI]$ is expected value of random clusterings.

## H UMAP Weights

UMAP (McInnes et al., 2018) attempts to estimate the density by comparing the distance to the k-nearest neighbors as follows:

$$\rho_i = min\{d(x_i, x_{i_j})|1 \leq j \leq k, d(x_i, x_{i_j}) > 0\}$$

$$\sum_{j=1}^{k} \exp(\frac{-max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}) = \log_2(k)$$

Here, $d(x_i, x_{i_j})$ is the distance between $x_i$ and $x_{i_j}$. $\rho_i$ is the minimum distance to $x_i$'s closest neighbor. $\sigma_i$, which smooths and normalizes the distances to the nearest neighbors, is calculated for each data point. Next, UMAP calculates the following weights between data points:

$$w((x_i, x_j)) = \exp(\frac{-max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i})$$

We use $1 - w((x_i, x_j))$ for agglomerative clustering.

## I Reproducibility

The SBERT model we used, along with the size of the $Q$ and $K$ layers use a dimension of size 384. Our total model has 34,839,937 parameters, of which 1,479,937 do not belong to SBERT. Input uses the ‘ldc_scope’ part of the ACE event mention. Our model takes roughly 2 hours to train on one V100 GPU, including the early stopping calculations which are done with the model set to ‘evaluation’ mode. We used batch size 10, which is the most that would fit in memory. For learning rates, we considered the suggestions in (Devlin et al., 2019). Datasets used are in English. ACE05 contains 5,349 mentions which fall into 33 event types. Data artifacts in this work were used for research purposes consistent with their licensing agreements and intended use. Artifacts we create (e.g. code and manual linking) are in line with this intended use.

## J Ablations

| Ablation | **Method** | | Clusters | Geometric NMI | Fowlkes Mallows | Completeness | Homogeneity | V-Measure | ARI |
|---|---|---|---|---|---|---|---|---|---|
| Original | Attn-Cosine | Agglo | 23 | 46.40 | 34.60 | 49.82 | 43.24 | 46.27 | 26.69 |
| | Attn-DotProduct | Agglo | 23 | 50.17 | 37.48 | 53.50 | 47.06 | 50.06 | 30.13 |
| | Attn | Manifold | 23 | 54.83 | 42.77 | 55.00 | 54.67 | 54.82 | 38.74 |
| | FT-SBERT | Manifold | 23 | 60.28 | 50.63 | 60.19 | 60.37 | 60.28 | 47.24 |
| | Attn-DotProduct | Affinity | 49-68 | 56.87 | 35.64 | 49.58 | 65.26 | 56.33 | 30.02 |
| | Attn-Cosine | Affinity | 50-69 | 56.54 | 33.00 | 48.72 | 65.62 | 55.91 | 27.04 |
| No Augmentation | Attn-Cosine | Agglo | 23 | 47.69 | 33.21 | 49.57 | 45.88 | 47.66 | 26.43 |
| | Attn-DotProduct | Agglo | 23 | 48.14 | 34.02 | 49.73 | 46.59 | 48.11 | 27.53 |
| | Attn | Manifold | 23 | 48.72 | 37.91 | 49.99 | 47.49 | 48.71 | 33.11 |
| | FT-SBERT | Manifold | 23 | 57.04 | 45.61 | 57.20 | 56.89 | 57.04 | 41.86 |
| | Attn-DotProduct | Affinity | 53 | 51.84 | 30.32 | 45.41 | 59.19 | 51.39 | 25.00 |
| | Attn-Cosine | Affinity | 56 | 52.88 | 31.93 | 45.83 | 61.02 | 52.35 | 26.19 |
| BERT Pooled Token Representation | Attn-Cosine | Agglo | 23 | 29.27 | 19.71 | 32.04 | 26.74 | 29.15 | 10.36 |
| | Attn-DotProduct | Agglo | 23 | 27.81 | 21.30 | 33.37 | 23.17 | 27.35 | 8.31 |
| | Attn | Manifold | 23 | 28.07 | 16.89 | 28.06 | 28.08 | 28.07 | 10.88 |
| | FT-BERT | Manifold | 23 | 29.72 | 16.17 | 28.95 | 30.52 | 29.71 | 10.89 |
| | Attn-DotProduct | Affinity | 23 | 27.99 | 16.26 | 27.89 | 28.09 | 27.99 | 10.41 |
| | Attn-Cosine | Affinity | | | | DNC | | | |
| BERT Trigger Representation<br>No Augmentation | Attn-Cosine | Agglo | 23 | 52.61 | 36.98 | 74.74 | 37.04 | 49.53 | 18.12 |
| | Attn-DotProduct | Agglo | 23 | 45.17 | 33.33 | 71.03 | 28.72 | 40.91 | 12.64 |
| | FT-BERT | Agglo | 23 | 51.67 | 35.05 | 76.18 | 35.04 | 48.00 | 14.92 |
| | Attn | Manifold | 23 | 60.018 | 44.09 | 73.70 | 48.88 | 58.77 | 31.21 |
| | FT-BERT | Manifold | 23 | 81.20 | 73.27 | 82.96 | 79.48 | 81.19 | 71.05 |
| | Attn-DotProduct | Affinity | 58 | 61.07 | 37.14 | 27.89 | 69.05 | 60.61 | 24.83 |
| | Attn-Cosine | Affinity | 32 | 61.97 | 37.02 | 68.50 | 56.06 | 61.66 | 25.40 |
| BERT Untrained | Untuned | Agglo | 23 | 61.19 | 45.36 | 70.73 | 52.94 | 60.56 | 34.51 |
| | Untuned | Manifold | 23 | 75.49 | 64.61 | 77.69 | 73.34 | 75.46 | 61.54 |
| RoBERTa-base Trigger Representation<br>No Augmentation | Attn-Cosine | Agglo | 23 | 49.37 | 33.48 | 69.54 | 35.05 | 46.61 | 13.57 |
| | Attn-DotProduct | Agglo | 23 | 43.62 | 31.33 | 66.27 | 28.71 | 40.06 | 10.24 |
| | FT-RoBERTa | Agglo | 23 | 57.32 | 36.68 | 75.69 | 43.41 | 55.18 | 18.17 |
| | Attn | Manifold | 23 | 57.14 | 44.30 | 69.72 | 46.84 | 56.03 | 33.01 |
| | FT-RoBERTa | Manifold | 23 | 83.44 | 78.12 | 84.37 | 82.52 | 83.44 | 76.56 |
| | Attn-DotProduct | Affinity | 44 | 57.76 | 39.86 | 64.06 | 52.08 | 57.45 | 29.17 |
| | Attn-Cosine | Affinity | 32 | 62.53 | 45.00 | 66.17 | 59.09 | 62.43 | 37.39 |
| RoBERTa-base Untrained | Untuned | Agglo | 23 | 17.53 | 23.66 | 39.62 | 7.76 | 12.98 | 0.35 |
| | Untuned | Manifold | 23 | 72.38 | 62.79 | 71.73 | 73.03 | 72.38 | 60.24 |
| ELECTRA-base Trigger Representation<br>No Augmentation | Attn-Cosine | Agglo | 23 | 21.07 | 23.36 | 37.36 | 11.89 | 18.03 | 1.36 |
| | Attn-DotProduct | Agglo | 23 | 21.52 | 23.11 | 36.23 | 12.78 | 18.89 | 1.63 |
| | FT-ELECTRA | Agglo | 23 | 30.02 | 25.74 | 47.36 | 19.03 | 27.15 | 4.87 |
| | Attn | Manifold | 23 | 32.45 | 28.89 | 54.54 | 19.30 | 28.51 | 8.67 |
| | FT-ELECTRA | Manifold | 23 | 62.41 | 47.05 | 63.92 | 60.93 | 62.39 | 42.65 |
| | Attn-DotProduct | Affinity | 278 | 45.76 | 22.35 | 44.28 | 47.29 | 45.74 | 12.35 |
| | Attn-Cosine | Affinity | 154 | 45.83 | 23.86 | 46.88 | 44.81 | 45.82 | 12.32 |
| ELECTRA-base Untrained | Untuned | Agglo | 23 | 12.82 | 21.14 | 22.91 | 7.17 | 10.93 | 0.49 |
| | Untuned | Manifold | 23 | 30.31 | 20.18 | 33.08 | 27.79 | 30.20 | 12.08 |
| ALBERT-base Trigger Representation<br>No Augmentation | Attn-Cosine | Agglo | 23 | 48.55 | 35.22 | 61.79 | 38.14 | 47.17 | 19.94 |
| | Attn-DotProduct | Agglo | 23 | 40.38 | 31.29 | 56.43 | 28.90 | 38.22 | 13.75 |
| | FT-ALBERT | Agglo | 23 | 38.41 | 27.64 | 59.44 | 24.82 | 35.02 | 6.34 |
| | Attn | Manifold | 23 | 56.80 | 43.26 | 70.72 | 45.61 | 55.46 | 30.69 |
| | FT-ALBERT | Manifold | 23 | 74.67 | 65.42 | 74.48 | 74.87 | 74.67 | 63.03 |
| | Attn-DotProduct | Affinity | 33 | 57.99 | 38.48 | 58.14 | 57.84 | 57.99 | 32.85 |
| | Attn-Cosine | Affinity | 40 | 60.27 | 34.81 | 60.01 | 60.53 | 60.27 | 26.71 |
| ALBERT-base Untrained | Untuned | Agglo | 23 | 47.00 | 31.21 | 54.09 | 40.83 | 46.53 | 21.29 |
| | Untuned | Manifold | 23 | 56.85 | 39.43 | 57.30 | 56.41 | 56.85 | 35.02 |

Table 8: New event type induction ablation results (%). The first 'Method' subcolumn is the input for clustering and the second is the clustering algorithm used. FT stands for finetuned. SBERT indicates the SBERT representations were used rather than our learned attentions (Attn). DNC indicates did not converge. Agglo is agglomerative clustering, Affinity is (Frey and Dueck, 2007), and Manifold is as described in Section 3.7.1. For affinity, each run can produce a slightly different number of clusters. BERT models use '*bert-base-uncased*'. The BERT pooled used a batch size of 8 because it takes up more VRAM. 15 epochs were used for each model. The BERT trigger representations use the ground truth triggers, which allows the duplicate problem to be avoided and helps clue in the network. The version trained with our method and without training is shown.

## K Experiments on MAVEN

To further evaluate our method, we conduct experiments on the Maven dataset (Wang et al., 2020), which contains 118,732 event mentions and 168 event types. This provides an excellent example of a large dataset which may have a number of unannotated event types (or event types which need to converted to finer-grained typing). Since this semi-supervised task has not been done on MAVEN, we select the most common 150 types as seen and the remaining 18 types as unseen. To adapt early stopping to such a large dataset, we adapt our early stopping by calculating the silhouette score every 250 steps, which is roughly the same computation as a single ACE2005 epoch. We train our method for 8 epochs with other hyperparameters the same as the ACE2005 experiments. For Manifold clustering, we select an arbitrarily large $k = 3,000$ for computational reasons.

We experiment on using trigger representations from the event mentions. Unfortunately, these are specified by the dataset in terms of the its tokenization scheme, so converting to BERT tokenization incurs a small error rate (for example, some words are tokenized as 'unk' in MAVEN). We find that these errors do not prevent the representation from performing well.

Results are shown in Table 9. They show that SBERT representations do not work well for the mentions in the MAVEN dataset (we believe this is due to differences in the scope of what is considered an event mention). Training with our method still improves clustering on the SBERT representations from 9.06 ARI to 10.51. However, the attention mechanisms learned are less effective along with our manifold clustering approach. The *bert-base-uncased* representations using the trigger words perform much better as expected. Additionally, our method, using both learned attention and trained trigger representations, increases performance significantly. For manifold, this is from 44.84 ARI to as high as 67.01. As noted in the conclusion, the interaction here between manifold approximation and large language model representations is an interesting future research direction for better understanding these results. It may also be interesting to think about how to obtain a desirable representation for starting the training of our model. Another example that would benefit from this is using a neural dense retriever to retrieve frozen embeddings in a retrieval large language model, such as (Borgeaud et al., 2021).

| Representation | Method | | Clusters | Geometric NMI | Fowlkes Mallows | Completeness | Homogeneity | V-Measure | ARI |
|---|---|---|---|---|---|---|---|---|---|
| SBERT | Untuned | Agglo | 18 | 33.33 | 18.98 | 37.25 | 29.81 | 33.12 | 9.06 |
| SBERT | Untuned | Manifold | 18 | 37.07 | 19.61 | 37.82 | 36.33 | 37.06 | 13.14 |
| *bert-base-uncased* | Untuned | Agglo | 18 | 66.95 | 42.46 | 75.14 | 59.66 | 66.51 | 32.47 |
| *bert-base-uncased* | Untuned | Manifold | 18 | 69.05 | 48.71 | 69.54 | 68.56 | 69.05 | 44.83 |
| SBERT | Attn-Cosine | Agglo | 18 | 30.14 | 14.08 | 30.82 | 29.48 | 30.13 | 6.86 |
| SBERT | Attn-DotProduct | Agglo | 18 | 29.92 | 13.90 | 30.06 | 29.77 | 29.92 | 7.49 |
| SBERT | Finetuned | Agglo | 18 | 34.23 | 16.48 | 34.05 | 34.41 | 34.22 | 10.51 |
| SBERT | Attn-Cosine | Manifold | 18 | 32.54 | 14.91 | 32.60 | 32.48 | 32.54 | 8.68 |
| SBERT | Finetuned | Manifold | 18 | 33.61 | 16.26 | 34.10 | 33.13 | 33.61 | 9.67 |
| *bert-base-uncased* | Attn-Cosine | Agglo | 18 | 73.77 | 60.25 | 75.01 | 72.56 | 73.76 | 57.11 |
| *bert-base-uncased* | Attn-DotProduct | Agglo | 18 | 73.52 | 58.19 | 73.84 | 73.21 | 73.52 | 55.07 |
| *bert-base-uncased* | Finetuned | Agglo | 18 | 74.54 | 59.67 | 78.41 | 70.86 | 74.44 | 55.30 |
| *bert-base-uncased* | Attn-Cosine | Manifold | 18 | 76.48 | 64.08 | 76.82 | 76.14 | 76.48 | 61.43 |
| *bert-base-uncased* | Finetuned | Manifold | 18 | **79.93** | **69.49** | **81.35** | **78.54** | **79.92** | **67.01** |
| *bert-base-uncased* | Attn-DotProduct | Affinity | 37 | 74.47 | 49.70 | 66.07 | 83.95 | 73.94 | 43.79 |
| *bert-base-uncased* | Attn-Cosine | Affinity | 41 | 74.37 | 48.54 | 65.20 | 84.83 | 73.73 | 42.14 |
| SBERT | Attn-DotProduct | Affinity | 40 | 40.87 | 11.53 | 35.48 | 47.08 | 40.47 | 6.81 |
| SBERT | Attn-Cosine | Affinity | 38 | 40.46 | 11.52 | 35.39 | 46.26 | 40.10 | 6.79 |

Table 9: New event type induction results (%) on MAVEN (Wang et al., 2020). The first two sections are baseline representations without our method. The *bert-base-uncased* representation uses trigger representations. The first subcolumn of 'method' is the input for clustering and the second is the clustering algorithm used. 'Finetuned' indicates the finetuned model representations from our method were used and 'untuned' indicates our method was not used. Attn-score indicates our learned attention scores were used and the method for computing these scores from the query and key vectors. E stands for ensemble. Agglo is agglomerative clustering, Affinity is (Frey and Dueck, 2007), and Manifold is as described in Section 3.7.1. For affinity, each run can produce a slightly different number of clusters.