Supervised Clustering Loss for Clustering-Friendly Sentence Embeddings: an Application to Intent Clustering

Anonymous ACL submission

Abstract

Modern virtual assistants are trained to classify customer requests into a taxonomy of predesigned intents. Requests that fall outside of this taxonomy, however, are often unhandled and need to be clustered to define new experiences. Recently, state-of-the-art results in intent clustering were achieved by training a neural network with a latent structured prediction loss. Unfortunately, though, this new approach suffers from a quadratic bottleneck as it requires to compute a joint embedding representation for all pairs of utterances to cluster. To overcome this limitation, we instead cast the problem into a representation learning task, and we adapt the latent structured prediction loss to fine-tune sentence encoders, thus making it possible to obtain clustering-friendly single-sentence embeddings. Our experiments show that the supervised clustering loss returns state-of-the-art results in terms of clustering accuracy and adjusted mutual information.

1 Introduction

Most virtual assistants, like Alexa, Cortana, Google Home, and Siri, have a Natural Language Understanding (NLU) component that categorizes customers' requests into supported experiences, organized by domains and intents. However, when user requests don't fit into these categories, NLU models can fail, causing friction in human-machine interaction. Analyzing these out-of-scope utterances can help expand the assistant's capabilities, but manually inspecting all failing utterances is unfeasible. Therefore, automation is needed, such as clustering frictional utterances into new required experiences. This approach is valuable for expanding the assistants' capabilities in a user-driven way.

One way is to use pre-trained sentence embeddings with unsupervised clustering algorithms. Another option is to train a clustering model in a supervised manner using utterances with known intents. This supervised approach has been successful in co-reference resolution (Finley and Joachims, 2005) and has been recently applied to intent clustering. A seminal work by Haponchyk et al. (2018) uses measures of utterance similarity as input to either Latent Structural Support Vector Machines (LSSVM) or to a Latent Structured Perceptron (LSP) (Yu and Joachims, 2009; Fernandes et al., 2014). The same two algorithms - LSSVM and LSP - were later used by Haponchyk and Moschitti (2021) to train a fully Neural Supervised Clustering architecture (NSC) with utterances encoded through pre-trained large language models - e.g. BERT (Devlin et al., 2019). Supervised clustering techniques use graph structures to represent clusters and are highly effective, but have a quadratic complexity due to the need for edge weights between all possible sample pairs. In the NSC case, for example, all pairs of utterances must pass through a Convolutional Neural Network at both training- and inference-time.

To avoid this, we propose using the supervised clustering loss to fine-tune sentence encoders, producing clustering-friendly single-sentence embeddings. This turns supervised clustering into a metric or representation learning problem where we force embeddings to be globally more suitable for intent clustering. Our approach has the advantage of scaling linearly with the number of samples, as embeddings only need to be computed for all utterances, not all pairs. To validate our approach, we perform experiments on CLINC150 (Larson et al., 2019), BANKING77 (Casanueva et al., 2020), DSTC11 (Galley et al., 2022), HUW64 (Liu et al., 2021) and Massive (FitzGerald et al., 2022): these are 5 public benchmark datasets for intent clustering, both monolingual and multilingual. For each dataset we fine-tune mBERT (Devlin et al., 2019), XLM roBERTa (Conneau et al., 2020) and two state-of-the-art sentence encoders (All Mpnet Base and Paraphrase Multilingual Mpnet) with either our supervised clustering loss or one among cross entropy loss, cosine similarity loss, contrastive loss or triplet margin loss. Results show that, regardless of base sentence encoder or algorithm chosen to perform clustering, our proposed fine-tuning strategy induces state-of-the-art embeddings that perform equally or better than those obtained with all other tested metric learning losses, when evaluated on the intent clustering task. Our code has been attached to this submission and will be publicly released upon acceptance.

2 Related Works

This work lies at the intersection of three research areas: intent clustering, sentence embeddings, and structured prediction loss - which we will briefly review below.

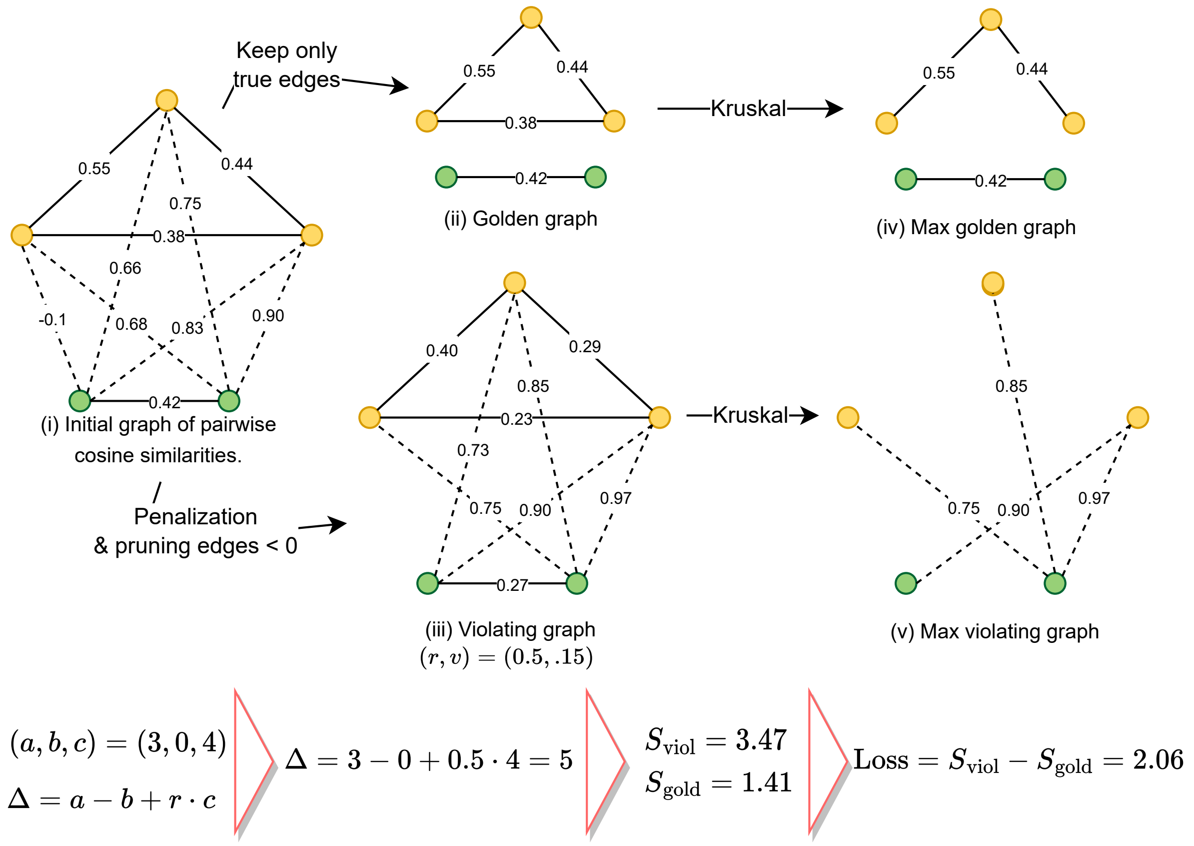

Figure 1: A sample calculation of the supervised clustering loss on two clusters (yellow points vs green points)

2.1 Intent Clustering

During the past few years, intent clustering has been a very active research topic. While it has been shown that pre-trained transformers perform poorly on out-of-scope detection (Zhang et al., 2022a), fine-tuning in a contrastive or semi-supervised fashion has proven beneficial (Casanueva et al., 2020; Zhang et al., 2021c; Mehri and Eric, 2021; Zhang et al., 2021d; Mou et al., 2022). Early works mostly focus on unsupervised clustering methods (Shi et al., 2018; Perkins and Yang, 2019; Chatterjee and Sengupta, 2020), but semi-supervision has now gained popularity (Forman et al., 2015; Zhang et al., 2022b). Lin et al. (2020), for example, propose to first perform supervised training on known intents and then use pseudo-labeling on unlabeled utterances to learn a better embedding space. Quite similarly, and in line with Deep Clustering (Caron et al., 2018), Zhang et al. (2021b) propose to first pre-train on known intents and then perform k-means clustering to assign pseudo-labels on unlabeled data. Finally, a structured prediction loss was used to directly teach both support vector machines (Finley and Joachims, 2005; Haponchyk et al., 2018) and neural networks (Haponchyk and Moschitti, 2021) to directly output intent clusters for some input utterances. This latter thread of research is the starting point of our work.

2.2 Sentence Embeddings

Current state-of-the-art sentence embeddings (Reimers and Gurevych, 2019, 2021; Liao, 2021; Kim et al., 2021; Giorgi et al., 2021) usually fine-tune pre-trained BERT-based architectures on SNLI (Bowman et al., 2015) and Multi-NLI (Williams et al., 2018) data with either a cross entropy loss, a contrastive loss or a triplet margin loss. Gao et al. (2021) and Yan et al. (2021) precisely show that contrastive loss can avoid an anisotropic embedding space. As for intent-friendly word and sentence embeddings, some works propose to pre-train BERT on open domain dialogs in a self-supervised manner (Mehri et al., 2020; Wu et al., 2020; Henderson et al., 2020; Hosseini-Asl et al., 2020). On the other hand, Zhang et al. (2020) formulated intent recognition as a sentence similarity task. Another common option consists in pre-training with a contrastive loss on intent detection tasks (Vulić et al., 2021; Zhang et al., 2021d). Finally and more generally, Zhang et al. (2021a) show that combining a contrastive loss with a clustering objective can improve short text clustering.

2.3 Structured Prediction

While in optimization problems local solutions often produce optimal results, structured prediction represents a valid alternative to solve NLP tasks requiring complex output, such as syntactic parsing (Roth and Yih, 2004), co-reference resolution (Yu and Joachims, 2009; Fernandes et al., 2014), and clustering (Finley and Joachims, 2005; Haponchyk et al., 2018). Nonetheless, relatively few works extend structured prediction theory to deep learning (LeCun et al., 2006; Durrett and Klein, 2015; Weiss et al., 2015; Kiperwasser and Goldberg, 2016; Peng et al., 2018; Milidiú and Rocha, 2018; Xu et al., 2018; Wang et al., 2019). In particular, when it comes to clustering, designing a differentiable loss function that captures the global characteristics of good clustering is particularly hard; for this reason, when dealing with coreference resolution - a closely related task - Lee et al. (2017) use simple losses, which already perform well but do not strictly take into account the cluster structure. Haponchyk and Moschitti (2021), on the other hand, represent clusters using graph structures and use LSSVM (Yu and Joachims, 2009) and LSP (Fernandes et al., 2014) - two structured prediction algorithms - to compute an augmented loss for training a deep clustering architecture.

3 Supervised Clustering Loss for Clustering-Friendly Representation Learning

In this section, we demonstrate how a structured learning approach - which utilizes *latent representations of graph structures* for predicting clusters from a set of utterances - can be instead used to fine-tune sentence encoders to be more clustering-friendly. Our approach is unique in that it leverages supervised clustering principles for the fine-tuning of sentence-transformers using examples of clusters, known as *gold clusters*. This allows for the creation of "cluster-friendly" embeddings, whose cosine similarities can be used to directly cluster the embedded utterances using various clustering algorithms such as threshold-based, K-Means, or Hierarchical Clustering.

Our fine-tuning loss represents utterances as nodes of a *fully-connected weighted graph*. The edge weights correspond to the cosine similarities between connected pairs of utterances (as defined by Eq. 2). By pruning the edges whose weight is below a certain threshold (i.e., the cosine similarity is less than 0), we can obtain a clustering. This clustering, however, is only used at training time to compute a clustering-sensitive loss, whose back-propagation contributes to the creation of more clustering-friendly sentence embeddings.

We begin by briefly explaining how we can leverage a supervised clustering loss to fine-tune sentence encoders, followed by a detailed description of the mathematical computation behind the loss.

3.1 Intuitive explanation of the Supervised Clustering Loss

Our loss function is inspired by the *Neural Supervised Clustering* (NSC) (Haponchyk and Moschitti, 2021). Specifically, the computation of the loss accounts for the differences between the golden clustering and the embedding-based clustering. The loss is made up of two components: a difference between two *scores* based on edge weights (Eqs. 9, 10), and a *structural-loss* based edge comparison (Eq. 8). Following the example in Figure 1:

1. at each learning step, we use the actual embeddings to compute a similarity matrix for the current clustering scenario, represented as a fully-connected graph (i);
2. using the golden clustering, we construct a first graph, called *gold graph* (ii), keeping only edges that connect nodes in the same clusters and pruning the others; its connected components now represent the golden clusters;
3. we construct a second graph, called *violating graph* (iii), perturbing the similarity matrix (i) by penalizing the edges connecting nodes in the same clusters; in this context, v is a real number between 0 and 1, representing the penalization factor on gold edges, while r represent what percentage of this penalization is transferred onto wrong edges;
4. we prune all the edges with weight below 0, resulting in a disconnected graph (iii), whose connected components are the predicted clusters;
5. to perform the comparison between the two resulting clusterings, we keep the minimum possible connectivity which preserves the connected components and select the strongest edges by applying Kruskal's Maximum Spanning Tree to each connected components, resulting in graphs (iv) and (v);
6. we compute a score for each graph - as the weight sum of the remaining edges, and the structural loss - as the difference between the number of edges of the golden graph and the numbers of correct and incorrect edges of the max-violating graph.
7. finally, we perform back propagation only in case the structural loss is greater than zero (which happens in the case of imperfect matching between the two graphs).

3.2 Algorithm details

Let $\{(x_i, y_i)\}_{i=1}^n$ be a set of samples to be clustered, where x_i represents the i-th object and y_i its cluster assignment. Let's further assume that $\text{Net}_\theta(.)$ is a generic neural network that encodes the objects $\{x_i\}_{i=1}^n$ into $k-$dimensional real-valued vectors, such that:

$$A = [\hat{x}_1, ..., \hat{x}_n] = \text{Net}_\theta([x_1, ..., x_n]), \quad (1)$$

where $A \in \mathbb{R}^{n \times k}$ contains all the n objects encoded with $\text{Net}_\theta(.)$.

The first step to compute the supervised clustering loss is to represent the clustering scenario $\{(x_i, y_i)\}_{i=1}^n$ through an undirected weighted graph, where the i-th node corresponds to x_i and the edge $e_{ij} = cosine\_similarity(\hat{x}_i, \hat{x}_j)$. In practice, the weighted adjacency matrix S with the pairwise cosine similarities fully defines the aforementioned graph. S can be efficiently computed through matrix multiplication in the following way:

$$S = 1 - \frac{\bar{A}\bar{A}^T}{2}, \tag{2}$$

where $\bar{A}$ is just the l_2-normalized version of A. Now, let D and $\bar{D}$ be two (n,n)-dimensional matrices such that:

$$D_{ij} = \begin{cases} 1 & \text{if } y_i = y_j \\ 0 & \text{otherwise} \end{cases} \quad \bar{D}_{ij} = \begin{cases} 1 & \text{if } y_i \neq y_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In other words, D is a mask for all the edges connecting any two samples sharing the same cluster (positive edges from now on), while $\bar{D}$ does the same for all the edges connecting any two samples in different clusters (negative edges from now on).

We will now define two graphs through their respective weighted adjacency matrices: i. a gold one where only positive edges are kept, and ii. a violating one, where weights on positive edges are decreased while weights on negative edges are increased.

$$S^{gold} = S \circ D \tag{4}$$

$$S^{viol} = max(0, S + v \cdot (r \cdot \bar{D} - D)) \tag{5}$$

In both equations, all operations are element-wise - for instance $S^{viol}_{ij} = \max(0, S_{ij} + v \cdot (r \cdot \bar{D}_{ij} - D_{ij}))$. The parameters $v, r \in \mathbb{R}^+$ are fine-tunable. They are meant to perturb the similarity matrix to make the edge selection for the correct clusters more challenging and more robust to fluctuation; v controls the impact of this pertubation, while r is used to unbalance the importance between positive and negative edges. On the possibly fully connected graph S^{viol}, we define clusters as the connected components obtained after neglecting all the edges, whose weights are less than a threshold τ. The next step is to exploit Kruskal's algorithm to compute the maximum spanning forest for both graphs.

$$H^{gold} = MaxSpanningForest(S^{gold}) \tag{6}$$

$$H^{viol} = MaxSpanningForest(S^{viol}) \tag{7}$$

In other words, H^{gold} and H^{viol} are two (n,n)-dimensional matrices whose elements are equal to 1 if the edge e_{ij} is included in the maximum spanning forest for S^{gold} and S^{viol} respectively. Intuitively, the nodes appearing in the same connected component in H are considered part of the same cluster.

H^{gold} results having the same clusters as D (i.e., the golden clusters), but D's connected components are fully-connected, whereas H^{gold}'s are minimally connected by virtue of Kruskal's algorithm (for a subgraph of n nodes, it has just $n-1$ edges, instead of the fully-connected n^2).

We are now ready to compute the loss. Let's first define some additional quantities: $a = sum(H^{gold})$, $b = sum(D \circ H^{viol})$ and $c = sum(\bar{D} \circ H^{viol})$ - where a is equal to the number of edges included in the maximum spanning forest on S^{gold}, while b is equal to the number of positive edges included in H^{viol}, and c to the number of negative edges included in H^{viol}. These three values are combined into a delta whose value decreases as more positive edges are included into the violating forest and increases when more negative ones are added:

$$\Delta = a - b + r \cdot c \tag{8}$$

Finally, let's compute two intermediate scores:

$$s_{gold} = sum(S \circ H^{gold}) \tag{9}$$

$$s_{viol} = sum(S \circ H^{viol}), \tag{10}$$

where s_{gold} and s_{viol} represent the sum of all edge weights/cosine similarities of the maximum spanning forest on the gold and violating graphs respectively. The supervised clustering loss will then be equal to:

$$\mathcal{L} = \begin{cases} s_{viol} - s_{gold} & \text{if } \Delta > 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

A graphical sample calculation of the supervised clustering loss can be found in figure 1.

Remark that the gradient cannot flow though the Δ component, nonetheless it is influenced by it by virtue of the condition for which $\mathcal{L} = 0$ if $\Delta > 0$.

3.3 Time complexity of the Algorithm

The time complexity for the computation of the supervised clustering loss is $O(n^2 \log n)$, where n is the number of utterances (see Sec. C.1 in the Appendix). This is still more efficient than other losses commonly used for fine-tuning sentence embeddings. For example, the naive implementation of the triplet loss has $O(n^3)$ complexity (Murphy, 2022). However, our experiments have shown that training time is not a significant issue for either loss, as the stopping criterion is typically triggered after just a few epochs.

4 Baseline Metric Losses

Using the same notation as in section 3, we will now define four other very well-known losses that proved effective in fine-tuning sentence encoders (Liao, 2021; Reimers and Gurevych, 2019; Nicosia and Moschitti, 2017). We used these losses as strong baselines for comparing the performance of our supervised clustering loss. Unlike the supervised clustering loss, these losses work on pairs or triplets of items and try to reorganise the embedding space simply by pushing away samples not sharing the same label while pulling closer those that do.

Let then $(\hat{x}_i, \hat{x}_j)$ be any two samples encoded with $Net_\theta(.)$ into $k-$dimensional real-valued vectors, and (y_i, y_j) their respective cluster assignments. We will define the Binary Classification Loss as:

$$\begin{cases} ln(\sigma(W(x_i, x_y, |x_i - x_y|))) & \text{if } y_i = y_j \\ 1 - ln(\sigma(W(x_i, x_y, |x_i - x_y|))) & \text{otherwise} \end{cases} \tag{12}$$

where $W(x_i, x_y, |x_i - x_y|)$ is just a linear projection applied to the concatenation of the two embeddings and their distance. Using instead the cosine similarity between x_i and x_j we can define the Cosine Similarity Loss as:

$$\begin{cases} [1 - cos\_sim(x_i, x_j)]^2 & \text{if } y_i = y_j \\ cos\_sim(x_i, x_j)^2 & \text{otherwise} \end{cases} \tag{13}$$

where the embeddings of samples sharing the same cluster are forced to have cosine similarity close to 1, while keeping the embeddings of non-related samples further apart. On the same line, the Contrastive Loss (Hadsell et al., 2006) can be defined as:

$$\begin{cases} cos\_dist(x_i, y_j)^2 & \text{if } y_i = y_j \\ max[0, m - cos\_dist(x_i, y_j)]^2 & \text{otherwise} \end{cases} \tag{14}$$

in this case, we force the embeddings of samples inside the same cluster to have cosine distance equal to zero, while keeping the cosine distance of non-related utterances above the margin m.

To conclude, we will present the Triplet Margin Loss which takes as input triples of samples $(\hat{x}_i, \hat{x}_j, \hat{x}_x)$ such that $y_i = y_j \neq y_x$ - where the first element is called the anchor, while the second and the third are commonly referred to as the positive and negative examples. The core idea behind this loss is to adjust the relative distances among the samples in each training triplet by minimizing the following quantity:

$$max[0, cos\_dist(x_i, y_j) - cos\_dist(x_i, x_z) - m] \tag{15}$$

in short, for all triplets, we want to cosine distance between the anchor and the negative to be higher than the distance between the anchor and the positive by at least the margin m.

5 Batch Sampling and Training Procedure

To fine-tune sentence embeddings, the training set plays a crucial role. The losses used for fine-tuning require specific samples to be manually engineered. The supervised clustering loss needs a 'clustering scenario' as input, while the other losses require pairs or triplets of samples with labels equal to 1 if they share the same cluster and 0 otherwise. To train, a common procedure involves randomly selecting k clusters from the training set and then randomly sampling m representatives from each cluster to form a training batch. A training epoch consists of n training batches.

For check-pointing and the stopping criterion, the Precision Recall Area Under the Curve (PRAUC) is monitored on pairs of utterances from the development set. At each training step, $m * k$ utterances are randomly sampled from the development set to calculate the cosine similarity among the sentence embeddings. At the end of each epoch, the PRAUC is computed using the true labels of pairs sharing the same cluster as 1 and pairs with different clusters as 0. This criterion ensures that the average cosine similarity between utterances with the same intent is higher than the average cosine similarity between utterances with different intents during training.

6 Experiments

In this section, we present experimental results on intent clustering using five losses applied to four sentence encoders, with resulting utterance embeddings clustered using Agglomerative Hierarchical Clustering. Appendix includes results from DBSCAN and a connected components-based procedure.

6.1 Benchmark Datasets

We experimented on five datasets commonly used for benchmarking intent classification and clustering: CLINC150 (Larson et al., 2019), BANKING77 (Casanueva et al., 2020), DSTC11 (Galley et al., 2022), HUW64 (Liu et al., 2021), and Massive (FitzGerald et al., 2022). The first four are in English, while Massive is multilingual and larger in size with almost 1 million manually translated utterances in 61 languages. To reduce its size, we randomly included 20% of the utterances. DSTC11 and BANKING77 are single-domain, while the rest are multi-domain. In essence, our study focuses on in-domain intent clustering. See Table 1 and Section A of the Appendix for dataset statistics and information on data acquisition and usage terms.

6.2 Base Models for Utterance Encoding

In our experiments, we rely on four different transformer-based sentence encoders and see whether our fine-tuning strategies improve their representation power when it comes to intent clustering:

1. Average pooling of the word-level **BERT** embeddings (Devlin et al., 2019). BERT was trained on the top 104 languages with the largest Wikipedia, using both a Masked Language Modeling (MLM) and a Next Sentence Prediction objectives,

2. Average pooling of the word-level **XLM roBERTa** embeddings (Conneau et al., 2020). XLM roBERTa is build on top of BERT but modifies key hyper-parameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates,

3. **All Mpnet Base** (Reimers and Gurevych, 2019) maps English-only sentences and paragraphs to a 768 dimensional dense vector space and was shown to be the best performing sentence encoder in English (HuggingFaceTeam, 2022). The model was trained on multiple corpora of sentence pairs using a Binary Classification Loss on top of a linear classifier that takes as input a concatenation of the two sentence embeddings,

4. **Paraphrase Multilingual Mpnet** (Reimers and Gurevych, 2020) maps sentences and paragraphs to a 768 dimensional dense vector space and was

DATASET	# domains	# intents	# languages	# total utterances	Avg utt. per intent	# train intent	# dev intent	# test intent
CLINC150	10	150	1 (en)	22500	150	90	30	30
DSTC11	1	22	1 (en)	2093	95	13	4	5
HWU64	21	64	1 (en)	11106	174	38	12	14
BANKING77	1	77	1 (en)	13242	172	46	15	16
Massive	18	60	51	759966	12666	30	22	16

Table 1: Intent Clustering Benchmark Dataset Statistics

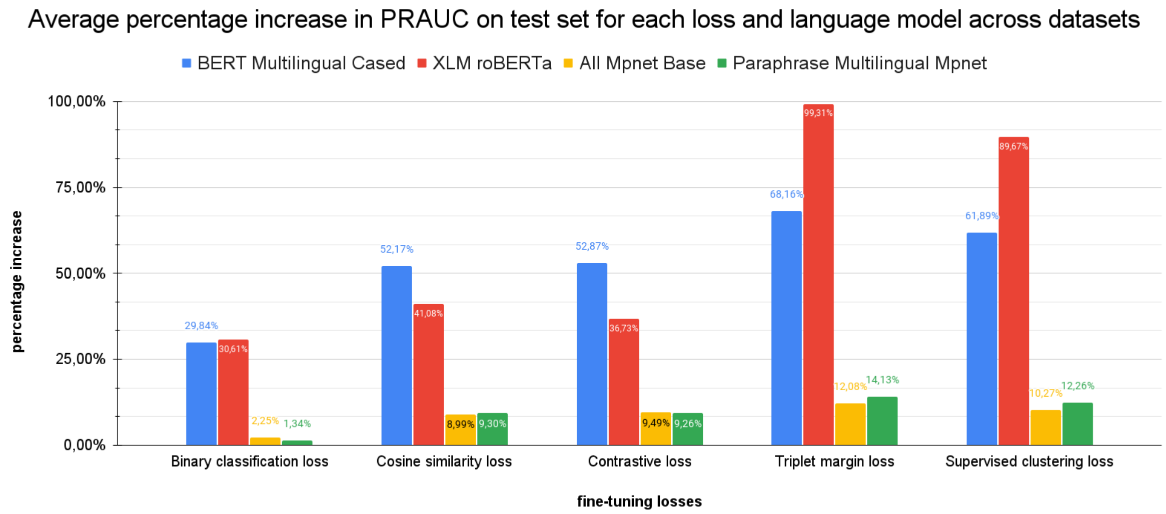

Figure 2: Fine-tuning always leads from moderate to large improvements in PRAUC on test utterances. The supervised clustering loss and the triplet margin loss clearly outperform all other losses. Increases on All Mpnet Base and Paraphrase Multilingual Mpnet are less pronounced because they were already on semantic similarity.

shown to be the best performing multilingual sentence encoder (HuggingFaceTeam, 2022). The model was trained on 1B sentence pairs using a Binary Classification Loss on top of the cosine similarity scores.

All Mpnet Base and Paraphrase Multilingual Mpnet Nonetheless were trained quite similarly to Sentence-BERT, but with more data.

6.3 Experimental setting

We randomly assign 60% of intents to the training set, 20% to the development set, and 20% to the test set for each of the 5 benchmark datasets. As detailed in section 5, the 4 base sentence encoders are separately fine-tuned using all training intent utterances and each of the five losses. Hyper-parameters are dataset-specific - see table 5 in the Appendix, and a max training epoch of 20 with 5 epochs of patience before early-stopping is set. The best parameters for the supervised clustering loss, triplet margin loss, and contrastive loss are selected via a grid search over specified intervals to obtain the highest PRAUC on the validation set. This procedure is repeated 5 times with different splits. The best parameters for the losses are stable across datasets and experiments: table 6 also shows the best values we used to obtain the final models. The final models consist of 20 fine-tuned models for each dataset (one per encoder-loss pair) except Massive, for which there are 15 fine-tuned models due to its multilingual nature. Information on hardware and computational cost can be found in section B of the Appendix.

Base and fine-tuned models are then used to extract embeddings for all the utterances in the development and test sets. After computing the matrix of pairwise cosine distances, we cluster utterances into tentative intents using agglomerative hierarchical clustering - an algorithm that recursively merges pairs of clusters based on a linkage criterion and a distance threshold. In the Appendix, we also report results using DBSCAN, and a procedure based on connected components. DBSCAN finds core samples of high density and expands clusters from them; in this case, the user needs to choose the minimum distance for two samples to be considered neighbors (ϵ) and the minimum number of samples around a candidate core sample. The third algorithm simply takes as clusters the connected components, after cutting all the edges below a certain threshold. The hyper-parameters of these three algorithms are optimized on the development set with respect to either the clustering accuracy or the adjusted mutual information score (AMIS). Table 7 in the Appendix contains the hyper-

DATASET	LOSS	BASE SENTENCE ENCODERS							
		BERT Multilingual Cased		XLM roBERTa		Paraphrase Multilingual Mpnet		All Mpnet Base	
		Average inter-intent pairwise cosine similarity	Average within-intent pairwise cosine similarity	Average inter-intent pairwise cosine similarity	Average within-intent pairwise cosine similarity	Average inter-intent pairwise cosine similarity	Average within-intent pairwise cosine similarity	Average inter-intent pairwise cosine similarity	Average within-intent pairwise cosine similarity
BANKING77	No fine-tuning	58,90%	67,10%	99,60%	99,70%	30,90%	58,30%	23,60%	56,00%
	Binary classification loss	21,20%	66,50%	99,60%	99,70%	31,50%	59,90%	27,80%	61,80%
	Cosine similarity loss	39,80%	69,90%	41,00%	68,40%	29,70%	72,10%	31,60%	72,80%
	Contrastive loss	32,70%	65,80%	32,80%	65,30%	22,50%	68,80%	23,20%	69,90%
	Triplet margin loss	25,80%	61,20%	48,70%	74,60%	16,40%	61,60%	13,80%	61,10%
	Supervised clustering loss	11,70%	39,70%	20,60%	54,10%	3,50%	45,10%	2,60%	44,90%
CLINC150	No fine-tuning	54,10%	67,50%	99,60%	99,70%	16,90%	61,70%	9,90%	53,10%
	Binary classification loss	50,00%	70,20%	99,50%	99,70%	17,40%	61,40%	10,90%	53,70%
	Cosine similarity loss	28,10%	78,20%	41,60%	71,10%	14,90%	79,60%	20,20%	77,40%
	Contrastive loss	20,80%	74,70%	22,80%	74,70%	8,70%	77,30%	15,80%	73,00%
	Triplet margin loss	21,00%	65,90%	37,30%	80,10%	5,40%	65,50%	6,30%	63,70%
	Supervised clustering loss	6,70%	44,10%	24,50%	63,40%	3,20%	50,50%	1,60%	49,50%
DSTC11	No fine-tuning	64,90%	69,90%	99,70%	99,70%	35,60%	62,20%	30,10%	57,80%
	Binary classification loss	34,90%	68,90%	-	-	-	-	24,50%	70,10%
	Cosine similarity loss	61,25%	79,35%	48,05%	78,10%	38,80%	77,95%	35,90%	75,50%
	Contrastive loss	27,60%	63,90%	45,20%	68,90%	24,50%	67,30%	28,10%	72,60%
	Triplet margin loss	34,90%	61,30%	47,05%	78,35%	12,80%	66,50%	12,80%	68,95%
	Supervised clustering loss	19,45%	49,30%	19,45%	63,15%	5,70%	55,80%	7,05%	58,30%
HWU64	No fine-tuning	47,90%	62,60%	99,40%	99,60%	16,00%	53,80%	11,10%	42,80%
	Binary classification loss	44,70%	65,90%	95,40%	97,90%	15,80%	54,10%	11,80%	42,90%
	Cosine similarity loss	38,30%	68,30%	98,30%	99,20%	22,40%	78,10%	16,40%	48,40%
	Contrastive loss	32,80%	65,80%	98,40%	99,20%	16,30%	75,60%	15,40%	76,70%
	Triplet margin loss	18,70%	69,90%	39,90%	79,80%	9,50%	59,20%	6,00%	57,10%
	Supervised clustering loss	6,20%	43,00%	97,40%	98,40%	1,70%	46,00%	1,50%	41,20%
Massive	No fine-tuning	41,60%	46,60%	99,30%	99,40%	22,80%	55,90%	-	-
	Binary classification loss	34,90%	63,50%	99,20%	99,30%	19,10%	53,40%	-	-
	Cosine similarity loss	40,60%	64,40%	98,70%	98,80%	31,00%	66,70%	-	-
	Contrastive loss	30,60%	62,90%	98,70%	98,20%	22,30%	62,90%	-	-
	Triplet margin loss	34,50%	61,50%	56,00%	77,30%	14,40%	54,70%	-	-
	Supervised clustering loss	8,70%	30,00%	20,30%	49,30%	2,50%	46,40%	-	-

Table 2: Pre-fine-tuning and post-fine-tuning average inter-intent and within-intent pairwise similarity on test utterances. The gap between the average inter-intent and within-intent pairwise similarities increases for all datasets, losses and base sentence encoders. In other words, whatever loss we use, utterances that share the same intent get closer while drifting apart from utterances with different intents. Interestingly enough, the supervised clustering loss behaves in a markedly different manner, yes reducing the within-intent pair-wise similarity, but also leading the inter-intent pair-wise similarity very close to zero. This is equal to say that the supervised clustering loss induces a topological space which is different from the one created by the other losses.

parameter search spaces. Test utterances are eventually clustered using the best hyper-parameters and the same metrics are computed. For each dataset, the whole experimental procedure - from fine-tuning to clustering - is repeated 5 times with different seeds and splits and average results are reported with their variance.

6.4 Performance of Fine-Tuning Strategies

Figure 2 shows that fine-tuning always leads to moderate or large improvements in PRAUC on test utterances, regardless of the loss or base sentence encoder chosen. The supervised clustering loss and the triplet margin loss are especially effective fine-tuning strategies. All Mpnet Base and Paraphrase Multilingual Mpnet show less pronounced increases since they were already fine-tuned on sentence similarity tasks. Table 8 in the Appendix confirms these results when broken down by dataset. Table 2 shows that improvements in PRAUC are reflected in average inter-intent and within-intent pairwise similarities- which should be interpreted jointly. In an ideal scenario, a loss should push the within-intent average cosine similarity close to 1 and the inter-intent average cosine similarity to 0. Nonetheless, in our analysis, we show that things go differently.

The gap between the average inter-intent and within-intent pairwise similarities increases for all datasets, losses and base sentence encoders. In other words, whatever loss we use, utterances that share the same intent get closer while drifting apart from utterances with different intents. Interestingly enough, however, while most losses increase the average within-intent pairwise similarity, the supervised clustering loss behaves in a markedly different manner, yes reducing the within-intent pair-wise similarity, but also leading the inter-intent pair-wise similarity very close to zero. This is equal to say that the supervised clustering loss induces a topological space which is different from the one created by the other losses. This is further confirmed when looking at figures 3, 4, 5, 6, 7, 8 in the Appendix - which show the tSNE plots of the BANKING77 test utterances when XLM-RoBERTa is used as base sentence encoder.

6.5 New Intent Clustering Results

The results of experiments with agglomerative hierarchical clustering using different datasets, sentence encoders, and losses are shown in tables 3 and 4. Although we performed comparable experiments with DBSCAN and a procedure based on connected components (see the Appendix), for every dataset the highest clustering accuracy and adjusted mutual information score were achieved with agglomerative hierarchical clustering on embeddings obtained from one of the four sentence encoders, fine-tuned with either the supervised clustering loss or the triplet margin loss. Moreover, since the supervised clustering loss re-arranges the embedding space by retaining edges only among utterances sharing the same intent, embeddings obtained from any sentence encoder fine-tuned with such loss are expected to be particularly suitable for agglomerative hierarchical clustering.

As shown in table 3, when we optimize the clustering algorithm hyper-parameters with respect to the adjusted mutual information score, in 13 cases out of 19 the supervised clustering loss proved to induce more clustering friendly embeddings, resulting in higher clustering performance. As further shown in table 4, the clustering behaviour slightly changes when we optimize

Average adjusted mutual information score on test set for all combinations of datasets, base sentence encoders and clustering algorithms when optimizing wrt the adjusted mutual information score									
Clustering algorithm	Base sentence encoder	Dataset	No Fine-Tuning	Binary classification loss	Cosine similarity loss	Contrastive loss	Triplet margin loss	Supervised clustering loss	BEST LOSS
Agglomerative Hierarchical Clustering	BERT Multilingual Cased	BANKING77	0.53±0.02	0.55±0.05	0.67±0.03	0.66±0.04	0.76±0.03	**0.77±0.05**	**Supervised clustering loss**
		CLINC150	0.73±0.02	0.76±0.03	0.77±0.04	0.77±0.04	0.84±0.03	**0.85±0.02**	**Supervised clustering loss**
		DSTC11	0.29±0.05	0.52±0.1	0.47±0.14	0.5±0.1	0.6±0.06	**0.63±0.1**	**Supervised clustering loss**
		HWU64	0.61±0.02	0.63±0.02	0.67±0.04	0.67±0.04	**0.72±0.05**	**0.72±0.04**	Triplet & Supervised
		Massive	0.27±0.01	0.36±0.05	0.45±0.04	0.46±0.04	**0.51±0.04**	**0.51±0.06**	Triplet & **Supervised**
	Paraphrase Multilingual Mpnet	BANKING77	0.74±0.02	0.72±0.07	0.76±0.06	0.75±0.05	**0.83±0.02**	0.81±0.03	Triplet margin loss
		CLINC150	0.86±0.03	0.87±0.02	0.88±0.03	0.87±0.03	0.92±0.02	**0.93±0.01**	**Supervised clustering loss**
		DSTC11	0.52±0.15	0.36±0.34	0.65±0.08	0.72±0.06	0.73±0.11	**0.75±0.11**	**Supervised clustering loss**
		HWU64	0.79±0.05	0.76±0.01	0.79±0.03	0.79±0.01	0.79±0.04	**0.81±0.04**	**Supervised clustering loss**
		Massive	0.6±0.09	0.6±0.06	0.65±0.06	0.64±0.06	**0.71±0.06**	0.7±0.05	Triplet margin loss
	All Mpnet Base	BANKING77	0.84±0.01	0.83±0.01	0.83±0.02	0.83±0.03	**0.88±0.02**	0.86±0.02	Triplet margin loss
		CLINC150	0.91±0.02	0.9±0.02	0.92±0.02	0.92±0.02	**0.94±0.01**	**0.94±0.01**	Triplet & **Supervised**
		DSTC11	0.49±0.17	0.63±0.16	0.75±0.14	0.71±0.12	**0.78±0.11**	0.7±0.1	Triplet margin loss
		HWU64	0.81±0.05	0.81±0.05	0.79±0.03	0.8±0.01	0.79±0.05	**0.85±0.03**	**Supervised clustering loss**
	XLM roBERTa	BANKING77	0.48±0.01	0.6±0.04	0.66±0.06	0.66±0.04	0.73±0.06	**0.75±0.03**	**Supervised clustering loss**
		CLINC150	0.66±0.02	0.72±0.07	0.74±0.05	0.71±0.07	0.86±0.03	**0.86±0.01**	**Supervised clustering loss**
		DSTC11	0.28±0.02	0.42±0.0	0.53±0.04	0.53±0.04	**0.68±0.05**	0.65±0.1	Triplet margin loss
		HWU64	0.52±0.04	0.61±0.09	0.56±0.05	0.55±0.07	0.73±0.05	**0.77±0.04**	**Supervised clustering loss**
		Massive	0.2±0.01	0.28±0.12	0.23±0.11	0.19±0.02	0.51±0.06	**0.58±0.04**	**Supervised clustering loss**

Table 3: Average adjusted mutual information score on test set using agglomerative hierarchical clustering, for all combinations of datasets and base sentence encoders - when optimizing wrt the adjusted mutual information score

Average clustering accuracy on test set for all combinations of datasets, base sentence encoders and clustering algorithms when optimizing wrt the clustering accuracy									
Clustering algorithm	Base sentence encoder	Dataset	No Fine-Tuning	Binary classification loss	Cosine similarity loss	Contrastive loss	Triplet margin loss	Supervised clustering loss	BEST LOSS
Agglomerative Hierarchical Clustering	BERT Multilingual Cased	BANKING77	0.32±0.05	0.37±0.06	0.52±0.04	0.5±0.08	**0.62±0.05**	**0.62±0.08**	Triplet & **Supervised**
		CLINC150	0.56±0.06	0.53±0.05	0.56±0.04	0.57±0.06	0.68±0.03	**0.71±0.06**	**Supervised clustering loss**
		DSTC11	0.33±0.05	0.65±0.1	0.56±0.08	0.6±0.11	0.65±0.14	**0.73±0.1**	**Supervised clustering loss**
		HWU64	0.52±0.04	0.51±0.03	0.56±0.06	0.55±0.04	**0.59±0.06**	0.56±0.04	Triplet margin loss
		Massive	0.22±0.03	0.41±0.07	0.46±0.04	0.51±0.04	**0.55±0.07**	0.53±0.08	Triplet margin loss
	Paraphrase Multilingual Mpnet	BANKING77	0.62±0.06	0.56±0.08	0.64±0.06	0.62±0.03	**0.72±0.03**	0.69±0.06	Triplet margin loss
		CLINC150	0.65±0.07	0.65±0.04	0.7±0.08	0.69±0.08	0.79±0.05	**0.83±0.05**	**Supervised clustering loss**
		DSTC11	0.57±0.09	0.48±0.17	0.75±0.1	0.73±0.06	0.75±0.15	**0.77±0.09**	**Supervised clustering loss**
		HWU64	0.73±0.09	0.74±0.1	0.69±0.05	0.67±0.03	**0.75±0.07**	0.68±0.05	Triplet margin loss
		Massive	0.62±0.09	0.61±0.07	0.68±0.05	0.6±0.08	0.67±0.11	**0.73±0.08**	**Supervised clustering loss**
	All Mpnet Base	BANKING77	0.7±0.04	0.67±0.05	0.68±0.04	0.71±0.07	**0.78±0.04**	0.73±0.04	Triplet margin loss
		CLINC150	0.75±0.06	0.75±0.06	0.77±0.05	0.78±0.08	0.81±0.03	**0.82±0.04**	**Supervised clustering loss**
		DSTC11	0.56±0.12	0.67±0.09	0.78±0.16	**0.83±0.12**	0.78±0.14	0.77±0.14	Cosine similarity loss
		HWU64	0.7±0.11	0.69±0.09	0.67±0.05	0.67±0.08	0.74±0.05	**0.78±0.08**	**Supervised clustering loss**
	XLM roBERTa	BANKING77	0.32±0.02	0.41±0.03	0.52±0.04	0.5±0.05	0.59±0.08	**0.62±0.04**	**Supervised clustering loss**
		CLINC150	0.54±0.03	0.6±0.1	0.55±0.03	0.55±0.04	**0.71±0.06**	0.7±0.04	Triplet margin loss
		DSTC11	0.36±0.08	0.68±0.0	0.57±0.19	0.61±0.18	**0.74±0.08**	0.71±0.09	Triplet margin loss
		HWU64	0.42±0.02	0.52±0.12	0.37±0.02	0.44±0.11	0.65±0.08	**0.73±0.07**	**Supervised clustering loss**
		Massive	0.23±0.02	0.3±0.09	0.26±0.09	0.22±0.02	0.52±0.04	**0.61±0.04**	**Supervised clustering loss**

Table 4: Average clustering accuracy on test set using agglomerative hierarchical clustering, for all combinations of datasets and base sentence encoders - when optimizing wrt the clustering accuracy

with respect to the clustering accuracy, with the supervised clustering loss outperforming other losses in 11 out of 19 cases. Overall, the supervised clustering loss and the triplet margin loss tended to perform similarly and significantly better than other tested losses. However, in some cases, one loss outperformed the other by up to 8 percentage points in clustering accuracy or adjusted mutual information score, indicating that the best loss depends on both the dataset and the base language model chosen. Further investigation is warranted. Notably, even pre-trained sentence encoders benefited significantly from fine-tuning with either the supervised clustering loss or the triplet margin loss, underscoring the difference between intent similarity and semantic similarity.

7 Conclusions and Future Work

We proposed a supervised clustering loss to fine-tune sentence encoders, enabling the production of clustering-friendly sentence embeddings. These embeddings can be used with any unsupervised clustering algorithm to discover new intents, overcoming the quadratic bottleneck of current supervised clustering architectures. Extensive experiments on 5 benchmark datasets, including both monolingual and multilingual data, and 4 different base sentence encoders showed that our fine-tuning strategy induced embeddings that perform equally or better than those obtained with all other tested metric learning losses when comparing their performance on intent clustering. In the future, we plan to analyze the characteristics of the embedding spaces induced by different losses to understand why the supervised clustering loss works well with agglomerative hierarchical clustering but not with DBSCAN. Notably, regardless of the loss or sentence encoder chosen, fine-tuned embeddings always improve the performance of unsupervised intent clustering.

8 Limitations and Ethical Considerations

Our work suggests further research on unsupervised clustering algorithms, investigating the performance of sentence embeddings generated using different clustering algorithms and losses. Additionally, more exploration is needed on the structural and topological differences in embedding space between supervised clustering loss and other losses. Although our experiments demonstrate the effectiveness of supervised clustering loss, we acknowledge the need for further investigation into the circumstances in which triplet margin loss may be preferable. Finally, while we strive to consider less conventional requests, biases in clustering systems may lead to oversimplification of people's requests, and we welcome further research on addressing this issue.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Ajay Chatterjee and Shubhashis Sengupta. 2020. Intent mining from past conversations for conversational agent. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4140–4152, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. 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.

Greg Durrett and Dan Klein. 2015. Neural CRF parsing. 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 302–312, Beijing, China. Association for Computational Linguistics.

Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2014. Latent trees for coreference resolution. *Computational Linguistics*, 40(4):801–835.

Thomas Finley and Thorsten Joachims. 2005. Supervised clustering with support vector machines. In *Proceedings of the 22nd international conference on Machine learning*, pages 217–224.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

George Forman, Hila Nachlieli, and Renato Keshet. 2015. Clustering by intent: a semi-supervised method to discover relevant clusters incrementally. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 20–36. Springer.

Michel Galley, Yun-Nung Chen, Raghav Gupta, Zhang Chen, Paul Crook, Seungwhan Moon, Chulaka Gunasekara, Satwik Kottur, Sarik Ghazarian, and Behnam Hedayatnia. 2022. Dstc11: The eleventh dialog system technology challenge. `https://dstc11.dstc.community/`. Accessed: 2010-09-30.

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.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. 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 879–895, Online. Association for Computational Linguistics.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Iryna Haponchyk and Alessandro Moschitti. 2021. Supervised neural clustering via latent structured output learning: Application to question intents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3364–3374, Online. Association for Computational Linguistics.

Iryna Haponchyk, Antonio Uva, Seunghak Yu, Olga Uryupina, and Alessandro Moschitti. 2018. Supervised clustering of questions into intents for dialog system applications. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2310–2321, Brussels, Belgium. Association for Computational Linguistics.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

HuggingFaceTeam. 2022. Comparison of sentence encoder performances. `https://www.sbert.net/docs/pretrained_models.html`. Accessed: 2022-10-18.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. 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 2528–2540, Online. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. 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 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Danqi Liao. 2021. Sentence embeddings using supervised contrastive learning. *arXiv preprint arXiv:2106.04791*.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8360–8367.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction*, pages 165–183. Springer.

Shikib Mehri and Mihail Eric. 2021. Example-driven intent prediction with observers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2979–2992, Online. Association for Computational Linguistics.

Shikib Mehri, Mihail Eric, and Dilek Z. Hakkani-Tür. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *ArXiv*, abs/2009.13570.

Ruy Luiz Milidiú and Rafael Rocha. 2018. Structured prediction networks through latent cost learning. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 645–649.

Yutao Mou, Keqing He, Pei Wang, Yanan Wu, Jingang Wang, Wei Wu, and Weiran Xu. 2022. Watch the neighbors: A unified k-nearest neighbor contrastive learning framework for ood intent discovery. *arXiv preprint arXiv:2210.08909*.

Kevin P Murphy. 2022. *Probabilistic machine learning: an introduction*. MIT press.

Massimo Nicosia and Alessandro Moschitti. 2017. Accurate sentence matching with hybrid siamese networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 2235–2238, New York, NY, USA. Association for Computing Machinery.

Hao Peng, Sam Thomson, and Noah A. Smith. 2018. Backpropagating through structured argmax using a SPIGOT. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1863–1873, Melbourne, Australia. Association for Computational Linguistics.

Hugh Perkins and Yi Yang. 2019. Dialog intent induction with deep multi-view clustering. 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 4016–4025, Hong Kong, China. Association for Computational Linguistics.

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.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2021. The curse of dense low-dimensional information retrieval for large index sizes. 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 605–611, Online. Association for Computational Linguistics.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Chen Shi, Qi Chen, Lei Sha, Sujian Li, Xu Sun, Houfeng Wang, and Lintao Zhang. 2018. Auto-dialabel: Labeling dialogue data with unsupervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 684–689, Brussels, Belgium. Association for Computational Linguistics.

Ivan Vulić, Pei-Hao Su, Samuel Coope, Daniela Gerz, Paweł Budzianowski, Iñigo Casanueva, Nikola Mrkšić, and Tsung-Hsien Wen. 2021. ConvFiT: Conversational fine-tuning of pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1151–1168, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Po-Wei Wang, Priya Donti, Bryan Wilder, and Zico Kolter. 2019. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *International Conference on Machine Learning*, pages 6545–6554. PMLR.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. 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 323–333, Beijing, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. 2018. A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, pages 5502–5511. PMLR.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. 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 5065–5075, Online. Association for Computational Linguistics.

Chun-Nam John Yu and Thorsten Joachims. 2009. Learning structural svms with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1169–1176, New York, NY, USA. Association for Computing Machinery.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021a. 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.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021b. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14365–14373.

Haode Zhang, Yuwei Zhang, Li-Ming Zhan, Jiaxin Chen, Guangyuan Shi, Xiao-Ming Wu, and Albert Y.S. Lam. 2021c. Effectiveness of pre-training for few-shot intent classification. In *Findings of the*

Association for Computational Linguistics: EMNLP 2021, pages 1114–1120, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianguo Zhang, Trung Bui, Seunghyun Yoon, Xiang Chen, Zhiwei Liu, Congying Xia, Quan Hung Tran, Walter Chang, and Philip Yu. 2021d. Few-shot intent detection via contrastive pre-training and fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1906–1912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5064–5082, Online. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Yao Wan, Zhiwei Liu, Ye Liu, Caiming Xiong, and Philip Yu. 2022a. Are pre-trained transformers robust in intent classification? a missing ingredient in evaluation of out-of-scope intent detection. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 12–20, Dublin, Ireland. Association for Computational Linguistics.

Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2022b. New intent discovery with pre-training and contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 256–269, Dublin, Ireland. Association for Computational Linguistics.

A Dataset licenses and release

DSTC11, Massive and HUW64 datasets are licensed under the Apache-2.0 License, while CLINC150 and BANKING77 are released under the cc-by-4.0 Creative Commons Public Licence. Massive can be downloaded from `https://github.com/jianguoz/Few-Shot-Intent-Detection`, DSTC11 from `https://github.com/amazon-science/dstc11-track2-intent-induction` and all the other datasets from `https://github.com/jianguoz/Few-Shot-Intent-Detection`. None of the dataset contains any offensive content or information that names or uniquely identifies individual people. Finally, our code includes a pre-processing script for every dataset that allows to turn the downloaded files into the format required in our pipeline.

B Hardware Infrastructure and Computational Budget

We perform our experiments on one Amazon EC2 P3.16 instance, a 64-bit architecture with 488 GB of RAM, Intel Xeon E5-2686 v4 (64-core CPU running at 2.30GHz) and 8x Nvidia Tesla V100 Tensor Core GPUs with 128 GB of VRAM.

C Time Complexity

C.1 Supervised Clustering Loss

Assumomg V is the number of nodes (utterances), and E is the number of edges (all utterances pairs) in figure 1, the time complexity is $O(V^2 \log V)$.

This result is the sum of the complexities for the following steps:

1. Computation of the S similarity matrix (Eq. 2) has quadratic complexity $O(V^2)$.
2. Element-wise product (Eq. 4) and pairwise addition/subtraction (Eq. 5) have quadratic complexity $O(V^2)$.
3. Computing the maximum spanning forests (MSF) by Kruskal's algorithm (Eq. 6) and (Eq. 7) is $(E \log V)$. In our case, the gold MSF will be computed only on correct positive edges E^+, while the most-violating MSF will be computed on all the predicted positive edges E (both correct and incorrect). In the worst case, E is equal to all pairs of utterances V^2 (all nodes connected = all pairs of utterances classified as being similar). So, the resulting complexity is $O(V^2 \log V)$.
4. Computing the structural loss (Eq. 8) has $O(V)$ complexity. This is due to the fact that in the worst case scenario (i.e., a fully connected graph), Kruskal's algorithm would return $V - 1$ edges, resulting in a $O(V)$ complexity for both element-wise products and summations.
5. For the scores s_{gold} (Eq. 9) and s_{viol} (Eq. 10) the previous argument applies as well.
6. Computing the loss (Eq. 11) has $O(1)$ complexity.

Therefore, the overall complexity of the supervised clustering loss is $O(V^2 \log V)$.

C.2 Supervised Clustering predictions

After the system has been trained, the time complexity for prediction is $O(V'^2)$, where V' is the number of utterances to be clustered. This is due to the following steps:

1. Computation of the S similarity matrix (Eq. 2) has quadratic complexity $O(V'^2)$.
2. Computation of the connected components is linear in terms of the edges, hence has complexity $O(V'^2)$.

D Experiment Hyper-parameters

You can find here details of the experimented hyper-parameters of training datasets (Table 5), losses (Table 6), and clustering algorithms (Table 7).

E Fine-tuning complete experimental results

Please find below average PRAUC (Table 8) for pre-training and post-training on train, dev, and test sets for each dataset, loss, and base sentence encoder.

F Clustering complete experimental results

You can find here average clustering accuracy (Table 9) and adjusted mutual information score (Table 10) on test set for all combinations of datasets, base sentence encoders, and clustering algorithms.

G tSNE plots of test utterance embeddings

Figures 3, 4, 5, 6, 7, 8 show the tSNE plots of the BANKING77 test utterances when XLM-RoBERTa is used as base sentence encoder. All plots where obtained with the following hyper-parameters:

- Perplexity = 20
- Learning rate: 200
- Iterations: 2000

As shown in figure 3, when no fine-tuning is performed - the point cloud is scattered all around. Same thing happens when the binary classification loss is used to fine-tune the model. In contrast, after fine-tuning with the cosine similarity loss or with contrastive learning - figures 5 and 6, respectively - intents are much better separated. Such visual clustering further improves when the triplet margin loss or the supervised clustering loss are used as fine-tuning strategies - see figures 7 and 8.

DATASET	# intents per batch	# utterances per intent	# batches train epoch	# batches val epoch
CLINC150	30	5	5	5
DSTC11	4	30	4	2
HWU64	12	15	4	4
BANKING77	15	8	5	5
Massive	12	10	5	5

Table 5: Dataset-specific training hyper-parameters

LOSS	Hyper-parameters	Search space	Optimal values
Supervised Clustering Loss	c	([0,1]; step: 0.05)	0.15
	r	([0,1]; step: 0.05)	0.5
Triplet Margin Loss	m	([0,1]; step: 0.05)	0.15
Contrastive Loss	m	([0,2]; step: 0.10)	1.75
Binary Classification Loss	-	-	-
Cosine Similarity Loss	-	-	-

Table 6: Losses: hyper-parameter search spaces and optimal values

ALGORITHM	Hyper-parameters	Search space
Agglomerative Hierarchical Clustering	Linkage	ward, complete, average
	Distance Threshold	([0,1]; step: 0.05)
DBSCAN	Eps	([0,1]; step: 0.05)
	Min Samples	[2, 5, 10, 15, 20, 25, 30]
Connected components	Cut Threshold	([0,1]; step: 0.05)

Table 7: Clustering algorithms: hyper-parameters search spaces

DATASET	LOSS	BASE SENTENCE ENCODER																							
		BERT Multilingual Cased						XLM roBERTa						Paraphrase Multilingual Mpnet						All Mpnet Base					
		Pre Fine-Tuning PRAUC on Train Set	Post Fine-Tuning PRAUC on Train Set	Pre Fine-Tuning PRAUC on Dev Set	Post Fine-Tuning PRAUC on Dev Set	Pre Fine-Tuning PRAUC on Test Set	Post Fine-Tuning PRAUC on Test Set	Pre Fine-Tuning PRAUC on Train Set	Post Fine-Tuning PRAUC on Train Set	Pre Fine-Tuning PRAUC on Dev Set	Post Fine-Tuning PRAUC on Dev Set	Pre Fine-Tuning PRAUC on Test Set	Post Fine-Tuning PRAUC on Test Set	Pre Fine-Tuning PRAUC on Train Set	Post Fine-Tuning PRAUC on Train Set	Pre Fine-Tuning PRAUC on Dev Set	Post Fine-Tuning PRAUC on Dev Set	Pre Fine-Tuning PRAUC on Test Set	Post Fine-Tuning PRAUC on Test Set	Pre Fine-Tuning PRAUC on Train Set	Post Fine-Tuning PRAUC on Train Set	Pre Fine-Tuning PRAUC on Dev Set	Post Fine-Tuning PRAUC on Dev Set	Pre Fine-Tuning PRAUC on Test Set	Post Fine-Tuning PRAUC on Test Set
BANKING77	Binary classification loss	39,02%	55,40%	39,30%	47,85%	38,55%	46,32%	35,70%	56,80%	36,03%	54,07%	34,35%	49,32%	67,87%	71,13%	68,40%	68,32%	64,83%	66,42%	76,00%	78,02%	78,23%	80,45%	75,13%	76,25%
	Cosine similarity loss		71,93%		67,97%		61,50%		68,78%		63,58%		60,43%		82,97%		79,82%		73,95%		85,93%		85,37%		80,55%
	Contrastive loss		70,33%		67,03%		60,28%		69,00%		64,40%		61,28%		81,82%		80,00%		73,62%		86,27%		85,68%		80,67%
	Triplet margin loss		76,72%		70,53%		67,08%		79,10%		73,17%		70,53%		81,32%		78,08%		74,90%		85,72%		84,42%		80,33%
	Supervised clustering loss		81,68%		72,82%		68,93%		79,25%		73,12%		70,07%		85,88%		81,72%		78,10%		90,73%		85,85%		83,10%
CLINC150	Binary classification loss	53,48%	58,68%	52,38%	55,20%	52,72%	57,23%	46,67%	51,60%	46,83%	48,80%	47,33%	51,32%	82,53%	82,43%	81,60%	80,78%	80,95%	80,27%	85,95%	85,45%	85,57%	85,08%	84,78%	84,32%
	Cosine similarity loss		70,02%		66,00%		66,67%		63,25%		60,35%		58,32%		89,80%		84,00%		82,52%		94,27%		90,05%		89,60%
	Contrastive loss		71,00%		65,82%		66,62%		60,93%		58,07%		56,73%		90,48%		84,30%		83,13%		94,18%		90,80%		89,13%
	Triplet margin loss		82,85%		77,22%		76,60%		82,65%		76,65%		75,53%		93,00%		87,90%		87,63%		94,35%		90,90%		90,47%
	Supervised clustering loss		85,85%		76,38%		76,48%		83,57%		77,08%		75,52%		94,53%		88,02%		87,78%		96,42%		92,13%		91,47%
DSTC11	Binary classification loss	48,35%	83,40%	47,38%	77,20%	47,08%	76,78%	44,50%	69,30%	43,30%	55,80%	43,20%	70,60%	85,80%	92,25%	80,00%	78,40%	80,95%	81,25%	84,32%	88,63%	83,35%	90,15%	81,05%	88,63%
	Cosine similarity loss		90,77%		81,85%		82,90%		88,02%		82,67%		76,18%		95,18%		90,63%		90,72%		96,33%		93,25%		95,72%
	Contrastive loss		91,38%		82,22%		82,17%		85,83%		80,82%		76,23%		95,22%		90,78%		90,78%		96,58%		93,20%		94,88%
	Triplet margin loss		87,32%		82,20%		83,15%		87,68%		84,50%		83,57%		92,73%		88,98%		92,23%		94,20%		91,42%		93,23%
	Supervised clustering loss		95,53%		84,52%		86,52%		94,00%		87,30%		86,37%		95,78%		90,17%		92,12%		95,98%		93,50%		93,87%
HWU64	Binary classification loss	47,22%	53,02%	49,87%	55,07%	50,17%	53,45%	33,55%	49,22%	35,17%	48,33%	37,33%	49,03%	74,50%	76,95%	72,77%	73,48%	76,50%	75,25%	73,00%	71,98%	71,47%	71,83%	73,95%	72,75%
	Cosine similarity loss		69,77%		63,87%		62,28%		38,43%		36,17%		40,15%		86,52%		77,27%		78,87%		83,57%		77,58%		77,45%
	Contrastive loss		67,42%		64,12%		62,58%		38,82%		38,62%		40,53%		84,35%		76,63%		77,78%		85,70%		80,22%		79,82%
	Triplet margin loss		74,07%		67,30%		65,82%		59,52%		55,55%		58,70%		84,93%		78,50%		79,73%		87,82%		81,22%		82,65%
	Supervised clustering loss		82,55%		69,73%		69,53%		74,35%		67,73%		70,78%		89,83%		80,28%		81,60%		92,08%		82,63%		83,77%
Massive	Binary classification loss	30,97%	56,71%	30,57%	49,09%	29,06%	43,13%	26,52%	34,97%	26,17%	38,05%	24,78%	32,90%	63,93%	68,08%	65,73%	71,07%	62,48%	66,17%	-	-	-	-	-	-
	Cosine similarity loss		64,48%		53,60%		50,85%		31,87%		31,40%		29,63%		79,02%		74,05%		70,92%	-	-	-	-	-	-
	Contrastive loss		65,98%		54,57%		53,03%		25,37%		25,43%		24,73%		78,82%		74,55%		71,45%	-	-	-	-	-	-
	Triplet margin loss		63,75%		56,23%		53,00%		65,88%		60,32%		57,60%		79,28%		76,25%		73,33%	-	-	-	-	-	-
	Supervised clustering loss		70,57%		56,97%		56,38%		70,62%		63,57%		60,50%		80,07%		76,25%		74,15%	-	-	-	-	-	-

Table 8: Average pre-training and post-training PRAUC on train, dev and test sets for each dataset, loss and base sentence encoder. Regardless of the loss or base sentence encoder chosen, fine-tuning always leads from moderate to large improvements in PRAUC on test utterances. This is especially true when the supervised clustering loss or the triplet margin loss are used as fine-tuning strategies. In general, increases are much less pronounced on All Mpnet Base and Paraphrase Multilingual Mpnet since these two models were already fine-tuned on sentence similarity tasks and datasets.

Average clustering accuracy on test set for all combinations of datasets, base sentence encoders and clustering algorithms when optimizing wrt the clustering accuracy									
Clustering algorithm	Base sentence encoder	Dataset	No Fine-Tuning	Binary classification loss	Cosine similarity loss	Contrastive loss	Triplet margin loss	Supervised clustering loss	BEST LOSS
Agglomerative Hierarchical Clustering	BERT Multilingual Cased	BANKING77	0.32±0.05	0.37±0.06	0.52±0.04	0.5±0.08	0.62±0.05	0.62±0.08	**Supervised clustering loss**
		CLINC150	0.56±0.06	0.53±0.05	0.56±0.04	0.57±0.06	0.68±0.03	0.71±0.06	**Supervised clustering loss**
		DSTC11	0.33±0.05	0.65±0.1	0.56±0.08	0.6±0.11	0.65±0.14	0.73±0.1	**Supervised clustering loss**
		HWU64	0.52±0.04	0.51±0.03	0.56±0.06	0.55±0.04	0.59±0.06	0.56±0.04	Triplet margin loss
		Massive	0.22±0.03	0.41±0.07	0.46±0.04	0.51±0.04	0.55±0.07	0.53±0.08	Triplet margin loss
	Paraphrase Multilingual Mpnet	BANKING77	0.62±0.06	0.56±0.08	0.64±0.06	0.62±0.03	0.72±0.03	0.69±0.06	Triplet margin loss
		CLINC150	0.65±0.07	0.65±0.04	0.7±0.08	0.69±0.08	0.79±0.05	0.83±0.05	**Supervised clustering loss**
		DSTC11	0.57±0.09	0.48±0.17	0.75±0.1	0.73±0.06	0.75±0.15	0.77±0.09	**Supervised clustering loss**
		HWU64	0.73±0.09	0.74±0.1	0.69±0.05	0.67±0.03	0.75±0.07	0.68±0.05	Triplet margin loss
		Massive	0.62±0.09	0.61±0.07	0.68±0.05	0.6±0.08	0.67±0.11	0.73±0.08	**Supervised clustering loss**
	All Mpnet Base	BANKING77	0.7±0.04	0.67±0.05	0.68±0.04	0.71±0.07	0.78±0.04	0.73±0.04	Triplet margin loss
		CLINC150	0.75±0.06	0.75±0.06	0.77±0.05	0.78±0.08	0.81±0.03	0.82±0.04	**Supervised clustering loss**
		DSTC11	0.56±0.12	0.67±0.09	0.78±0.16	0.83±0.12	0.78±0.14	0.77±0.14	Cosine similarity loss
		HWU64	0.7±0.11	0.69±0.09	0.67±0.05	0.67±0.08	0.74±0.05	0.78±0.08	**Supervised clustering loss**
	XLM roBERTa	BANKING77	0.32±0.02	0.41±0.03	0.52±0.04	0.5±0.05	0.59±0.08	0.62±0.04	**Supervised clustering loss**
		CLINC150	0.54±0.03	0.6±0.1	0.55±0.03	0.55±0.04	0.71±0.06	0.7±0.04	Triplet margin loss
		DSTC11	0.36±0.08	0.68±0.0	0.57±0.19	0.61±0.18	0.74±0.08	0.71±0.09	Triplet margin loss
		HWU64	0.42±0.02	0.52±0.12	0.37±0.02	0.44±0.11	0.65±0.08	0.73±0.07	**Supervised clustering loss**
		Massive	0.23±0.02	0.3±0.09	0.26±0.09	0.22±0.02	0.52±0.04	0.61±0.04	**Supervised clustering loss**
Connected Components	BERT Multilingual Cased	BANKING77	0.13±0.02	0.17±0.04	0.39±0.09	0.36±0.08	0.43±0.09	0.46±0.1	**Supervised clustering loss**
		CLINC150	0.23±0.05	0.25±0.04	0.37±0.04	0.34±0.06	0.49±0.07	0.51±0.02	**Supervised clustering loss**
		DSTC11	0.4±0.11	0.38±0.11	0.52±0.08	0.49±0.1	0.54±0.12	0.46±0.11	Triplet margin loss
		HWU64	0.23±0.02	0.23±0.03	0.38±0.08	0.34±0.05	0.45±0.07	0.44±0.11	Triplet margin loss
		Massive	0.24±0.02	0.23±0.07	0.27±0.07	0.29±0.06	0.3±0.08	0.32±0.1	**Supervised clustering loss**
	Paraphrase Multilingual Mpnet	BANKING77	0.36±0.03	0.43±0.06	0.51±0.07	0.51±0.05	0.46±0.09	0.45±0.11	Cosine similarity loss
		CLINC150	0.51±0.06	0.49±0.08	0.58±0.12	0.57±0.1	0.66±0.08	0.63±0.02	Triplet margin loss
		DSTC11	0.44±0.12	0.66±0.09	0.69±0.12	0.7±0.08	0.71±0.13	0.72±0.08	**Supervised clustering loss**
		HWU64	0.48±0.09	0.5±0.09	0.48±0.18	0.52±0.16	0.57±0.11	0.53±0.08	Triplet margin loss
		Massive	0.35±0.06	0.39±0.1	0.44±0.05	0.41±0.05	0.41±0.05	0.39±0.08	Contrastive loss
	All Mpnet Base	BANKING77	0.48±0.04	0.5±0.06	0.49±0.08	0.55±0.09	0.55±0.05	0.54±0.06	Cosine similarity loss
		CLINC150	0.53±0.08	0.52±0.07	0.71±0.06	0.63±0.06	0.67±0.02	0.62±0.06	Contrastive loss
		DSTC11	0.39±0.14	0.67±0.13	0.67±0.12	0.67±0.1	0.77±0.1	0.73±0.1	Triplet margin loss
		HWU64	0.47±0.04	0.44±0.09	0.41±0.18	0.43±0.09	0.57±0.09	0.46±0.13	Triplet margin loss
	XLM roBERTa	BANKING77	0.08±0.0	0.11±0.04	0.35±0.07	0.33±0.05	0.39±0.08	0.51±0.07	**Supervised clustering loss**
		CLINC150	0.04±0.0	0.04±0.0	0.26±0.18	0.22±0.22	0.49±0.23	0.48±0.22	Triplet margin loss
		DSTC11	0.4±0.11	0.57±0.0	0.55±0.11	0.5±0.12	0.52±0.09	0.46±0.14	binary_classification
		HWU64	0.08±0.0	0.13±0.1	0.11±0.06	0.09±0.03	0.36±0.15	0.12±0.09	Triplet margin loss
		Massive	0.24±0.02	0.24±0.03	0.23±0.03	0.23±0.03	0.41±0.1	0.39±0.04	Triplet margin loss
DBSCAN	BERT Multilingual Cased	BANKING77	0.19±0.02	0.26±0.08	0.45±0.07	0.41±0.09	0.48±0.08	0.49±0.1	**Supervised clustering loss**
		CLINC150	0.25±0.05	0.28±0.04	0.38±0.07	0.4±0.07	0.54±0.05	0.53±0.02	Triplet margin loss
		DSTC11	0.39±0.1	0.52±0.06	0.59±0.12	0.5±0.11	0.54±0.17	0.57±0.1	Contrastive loss
		HWU64	0.29±0.05	0.37±0.06	0.42±0.09	0.46±0.06	0.51±0.05	0.44±0.09	Triplet margin loss
		Massive	0.25±0.03	0.39±0.08	0.43±0.07	0.47±0.08	0.5±0.07	0.45±0.05	Triplet margin loss
	Paraphrase Multilingual Mpnet	BANKING77	0.42±0.06	0.42±0.07	0.55±0.06	0.48±0.06	0.53±0.03	0.49±0.13	Contrastive loss
		CLINC150	0.5±0.09	0.5±0.07	0.57±0.13	0.61±0.1	0.65±0.1	0.64±0.03	Triplet margin loss
		DSTC11	0.56±0.07	0.69±0.06	0.7±0.13	0.78±0.07	0.64±0.06	0.71±0.07	Cosine similarity loss
		HWU64	0.52±0.12	0.5±0.15	0.61±0.12	0.59±0.14	0.66±0.1	0.53±0.04	Triplet margin loss
		Massive	0.5±0.05	0.45±0.09	0.52±0.05	0.55±0.08	0.47±0.09	0.54±0.07	Cosine similarity loss
	All Mpnet Base	BANKING77	0.47±0.05	0.49±0.07	0.58±0.08	0.57±0.09	0.61±0.06	0.55±0.08	Triplet margin loss
		CLINC150	0.5±0.05	0.54±0.08	0.71±0.07	0.67±0.04	0.67±0.02	0.64±0.05	Contrastive loss
		DSTC11	0.61±0.04	0.62±0.09	0.71±0.1	0.69±0.1	0.75±0.07	0.68±0.12	Triplet margin loss
		HWU64	0.41±0.07	0.47±0.11	0.53±0.14	0.58±0.11	0.62±0.09	0.52±0.11	Triplet margin loss
	XLM roBERTa	BANKING77	0.11±0.01	0.3±0.04	0.39±0.07	0.44±0.05	0.53±0.04	0.49±0.05	Triplet margin loss
		CLINC150	0.14±0.03	0.28±0.05	0.32±0.14	0.26±0.18	0.5±0.15	0.5±0.21	**Supervised clustering loss**
		DSTC11	0.39±0.1	0.41±0.0	0.58±0.1	0.56±0.13	0.52±0.07	0.59±0.09	**Supervised clustering loss**
		HWU64	0.24±0.02	0.28±0.05	0.19±0.05	0.24±0.16	0.58±0.1	0.39±0.06	Triplet margin loss
		Massive	0.23±0.02	0.27±0.05	0.27±0.04	0.24±0.03	0.52±0.05	0.47±0.06	Triplet margin loss

Table 9: Average clustering accuracy on test set for all combinations of datasets, base sentence encoders and clustering algorithms when optimizing wrt the clustering accuracy. It is worth mentioning that gaps in performance between the Supervised Clustering Loss and the Triplet Margin Loss are quite narrow, with confidence intervals often overlapping. On the contrary, all other losses clearly lag behind in terms of performance. Nevertheless, in all cases, fine-tuning any of the base sentence encoders with any of the losses proved beneficial - regardless of the dataset or clustering algorithm adopted.

Average adjusted mutual information score on test set for all combinations of datasets, base sentence encoders and clustering algorithms when optimizing wrt the adjusted mutual information score									
Clustering algorithm	Base sentence encoder	Dataset	No Fine-Tuning	Binary classification loss	Cosine similarity loss	Contrastive loss	Triplet margin loss	Supervised clustering loss	BEST LOSS
Agglomerative Hierarchical Clustering	BERT Multilingual Cased	BANKING77	0.53±0.02	0.55±0.05	0.67±0.03	0.66±0.04	0.76±0.03	0.77±0.05	**Supervised clustering loss**
		CLINC150	0.73±0.02	0.76±0.03	0.77±0.04	0.77±0.04	0.84±0.03	0.85±0.02	**Supervised clustering loss**
		DSTC11	0.29±0.05	0.52±0.1	0.47±0.14	0.5±0.1	0.6±0.06	0.63±0.1	**Supervised clustering loss**
		HWU64	0.61±0.02	0.63±0.02	0.67±0.04	0.67±0.04	0.72±0.05	0.72±0.04	Triplet margin loss
		Massive	0.27±0.01	0.36±0.05	0.45±0.04	0.46±0.04	0.51±0.04	0.51±0.06	**Supervised clustering loss**
	Paraphrase Multilingual Mpnet	BANKING77	0.74±0.02	0.72±0.07	0.76±0.06	0.75±0.05	0.83±0.02	0.81±0.03	Triplet margin loss
		CLINC150	0.86±0.03	0.87±0.02	0.88±0.03	0.87±0.03	0.92±0.02	0.93±0.01	**Supervised clustering loss**
		DSTC11	0.52±0.15	0.36±0.34	0.65±0.08	0.72±0.06	0.73±0.11	0.75±0.11	**Supervised clustering loss**
		HWU64	0.79±0.05	0.76±0.01	0.79±0.03	0.79±0.01	0.79±0.04	0.81±0.04	**Supervised clustering loss**
		Massive	0.6±0.09	0.6±0.06	0.65±0.06	0.64±0.06	0.71±0.06	0.7±0.05	Triplet margin loss
	All Mpnet Base	BANKING77	0.84±0.01	0.83±0.01	0.83±0.02	0.83±0.03	0.88±0.02	0.86±0.02	Triplet margin loss
		CLINC150	0.91±0.02	0.9±0.02	0.92±0.02	0.92±0.02	0.94±0.01	0.94±0.01	**Supervised clustering loss**
		DSTC11	0.49±0.17	0.63±0.16	0.75±0.14	0.71±0.12	0.78±0.11	0.7±0.1	Triplet margin loss
		HWU64	0.81±0.05	0.81±0.05	0.79±0.03	0.8±0.01	0.79±0.05	0.85±0.03	**Supervised clustering loss**
	XLM roBERTa	BANKING77	0.48±0.01	0.6±0.04	0.66±0.06	0.66±0.04	0.73±0.06	0.75±0.03	**Supervised clustering loss**
		CLINC150	0.66±0.02	0.72±0.07	0.74±0.05	0.71±0.07	0.86±0.03	0.86±0.01	**Supervised clustering loss**
		DSTC11	0.28±0.02	0.42±0.0	0.53±0.04	0.53±0.04	0.68±0.05	0.65±0.1	Triplet margin loss
		HWU64	0.52±0.04	0.61±0.09	0.56±0.05	0.55±0.07	0.73±0.05	0.77±0.04	**Supervised clustering loss**
		Massive	0.2±0.01	0.28±0.12	0.23±0.11	0.19±0.02	0.51±0.06	0.58±0.04	**Supervised clustering loss**
Connected Components	BERT Multilingual Cased	BANKING77	0.23±0.02	0.26±0.09	0.52±0.07	0.52±0.05	0.58±0.08	0.6±0.09	**Supervised clustering loss**
		CLINC150	0.38±0.04	0.43±0.05	0.51±0.15	0.58±0.06	0.72±0.04	0.72±0.03	Triplet margin loss
		DSTC11	0.13±0.03	0.37±0.13	0.43±0.12	0.44±0.15	0.5±0.11	0.33±0.15	Triplet margin loss
		HWU64	0.31±0.04	0.32±0.07	0.45±0.1	0.41±0.11	0.57±0.07	0.54±0.13	Triplet margin loss
		Massive	0.14±0.03	0.18±0.07	0.22±0.08	0.22±0.11	0.25±0.14	0.32±0.1	**Supervised clustering loss**
	Paraphrase Multilingual Mpnet	BANKING77	0.54±0.05	0.45±0.16	0.66±0.06	0.65±0.04	0.59±0.12	0.49±0.2	contrastive_learning
		CLINC150	0.69±0.05	0.7±0.06	0.72±0.18	0.76±0.08	0.83±0.04	0.78±0.08	Triplet margin loss
		DSTC11	0.37±0.12	0.58±0.02	0.6±0.15	0.61±0.11	0.65±0.12	0.65±0.11	**Supervised clustering loss**
		HWU64	0.55±0.1	0.52±0.1	0.58±0.18	0.57±0.13	0.64±0.12	0.62±0.08	Triplet margin loss
		Massive	0.32±0.08	0.33±0.15	0.45±0.08	0.39±0.13	0.41±0.06	0.4±0.08	contrastive_learning
	All Mpnet Base	BANKING77	0.59±0.07	0.67±0.04	0.69±0.07	0.71±0.06	0.69±0.02	0.63±0.13	Cosine similarity loss
		CLINC150	0.72±0.07	0.69±0.07	0.82±0.06	0.8±0.07	0.82±0.05	0.82±0.01	**Supervised clustering loss**
		DSTC11	0.19±0.17	0.5±0.11	0.47±0.29	0.55±0.22	0.65±0.17	0.67±0.13	**Supervised clustering loss**
		HWU64	0.49±0.14	0.46±0.15	0.5±0.15	0.62±0.11	0.68±0.08	0.62±0.07	Triplet margin loss
	XLM roBERTa	BANKING77	0.01±0.0	0.15±0.13	0.46±0.08	0.49±0.08	0.53±0.09	0.63±0.06	**Supervised clustering loss**
		CLINC150	0.0±0.0	0.0±0.0	0.37±0.31	0.28±0.34	0.62±0.31	0.63±0.32	**Supervised clustering loss**
		DSTC11	0.04±0.01	0.03±0.0	0.45±0.11	0.39±0.09	0.46±0.09	0.33±0.19	Triplet margin loss
		HWU64	0.0±0.0	0.08±0.16	0.09±0.18	0.03±0.06	0.51±0.26	0.08±0.16	Triplet margin loss
		Massive	0.0±0.0	0.08±0.1	0.04±0.08	0.0±0.0	0.34±0.09	0.36±0.08	**Supervised clustering loss**
DBSCAN	BERT Multilingual Cased	BANKING77	0.27±0.06	0.32±0.08	0.58±0.05	0.55±0.06	0.57±0.11	0.64±0.07	**Supervised clustering loss**
		CLINC150	0.4±0.04	0.45±0.05	0.58±0.08	0.57±0.08	0.71±0.05	0.71±0.04	**Supervised clustering loss**
		DSTC11	0.2±0.05	0.38±0.13	0.49±0.13	0.5±0.14	0.59±0.11	0.39±0.11	Triplet margin loss
		HWU64	0.41±0.04	0.49±0.06	0.55±0.1	0.61±0.04	0.58±0.13	0.57±0.11	Cosine similarity loss
		Massive	0.12±0.03	0.29±0.11	0.34±0.07	0.36±0.07	0.4±0.08	0.42±0.09	**Supervised clustering loss**
	Paraphrase Multilingual Mpnet	BANKING77	0.56±0.06	0.55±0.09	0.66±0.04	0.61±0.07	0.53±0.23	0.58±0.15	contrastive_learning
		CLINC150	0.64±0.1	0.71±0.05	0.77±0.08	0.69±0.24	0.82±0.04	0.8±0.03	Triplet margin loss
		DSTC11	0.39±0.18	0.63±0.04	0.71±0.09	0.75±0.06	0.71±0.07	0.72±0.09	Cosine similarity loss
		HWU64	0.52±0.14	0.59±0.09	0.66±0.2	0.71±0.09	0.75±0.07	0.66±0.05	Triplet margin loss
		Massive	0.44±0.08	0.52±0.13	0.51±0.06	0.5±0.05	0.46±0.13	0.57±0.06	**Supervised clustering loss**
	All Mpnet Base	BANKING77	0.63±0.03	0.66±0.03	0.66±0.08	0.65±0.09	0.64±0.14	0.63±0.16	contrastive_learning
		CLINC150	0.73±0.04	0.71±0.06	0.81±0.06	0.78±0.1	0.79±0.06	0.77±0.04	contrastive_learning
		DSTC11	0.5±0.13	0.56±0.2	0.71±0.09	0.73±0.07	0.76±0.1	0.73±0.11	Triplet margin loss
		HWU64	0.51±0.09	0.52±0.13	0.62±0.08	0.62±0.12	0.65±0.09	0.6±0.07	Triplet margin loss
	XLM roBERTa	BANKING77	0.03±0.02	0.42±0.08	0.5±0.13	0.52±0.1	0.66±0.03	0.65±0.06	Triplet margin loss
		CLINC150	0.21±0.04	0.45±0.06	0.46±0.2	0.39±0.24	0.7±0.13	0.67±0.25	Triplet margin loss
		DSTC11	0.06±0.04	0.24±0.0	0.49±0.14	0.47±0.11	0.56±0.13	0.46±0.16	Triplet margin loss
		HWU64	0.31±0.05	0.38±0.07	0.28±0.14	0.19±0.07	0.62±0.13	0.51±0.07	Triplet margin loss
		Massive	0.08±0.03	0.13±0.1	0.08±0.1	0.02±0.01	0.43±0.06	0.32±0.1	Triplet margin loss

Table 10: Average adjusted mutual information score on test set for all combinations of datasets, base sentence encoders and clustering algorithms when optimizing wrt the adjusted mutual information score. It is worth mentioning that gaps in performance between the Supervised Clustering Loss and the Triplet Margin Loss are quite narrow, with confidence intervals often overlapping. On the contrary, all other losses clearly lag behind in terms of performance. Nevertheless, in all cases, fine-tuning any of the base sentence encoders with any of the losses proved beneficial - regardless of the dataset or clustering algorithm adopted.

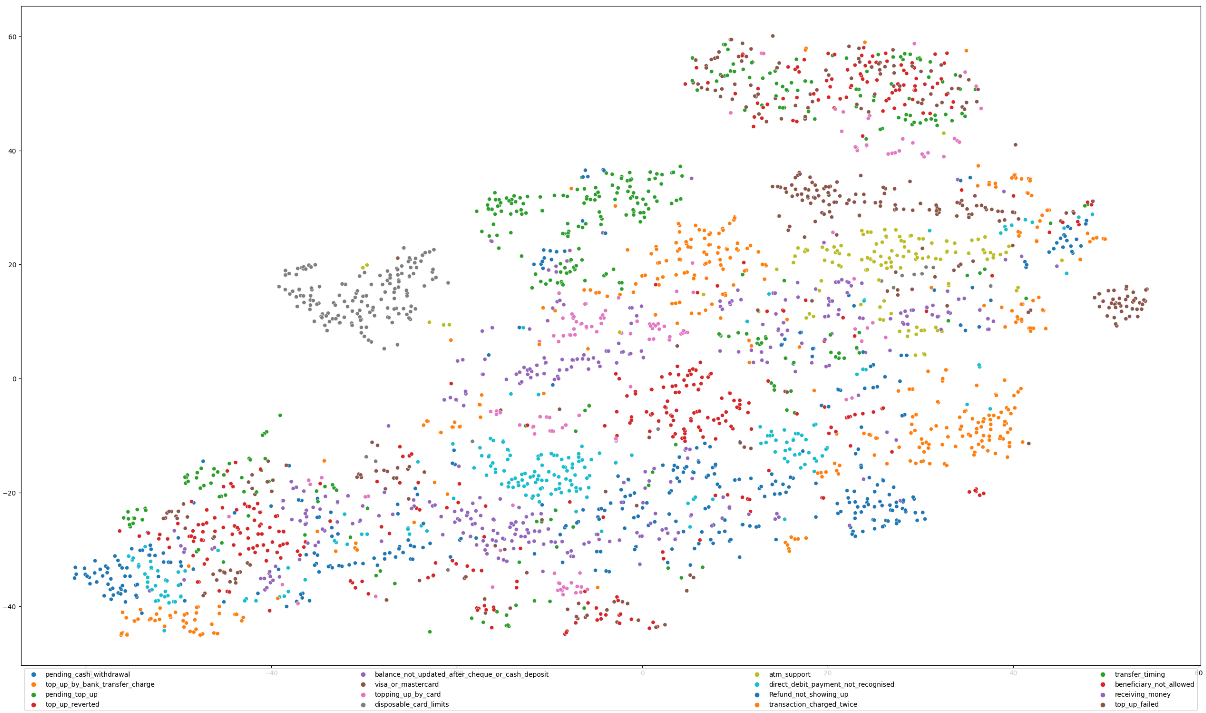

Figure 3: tSNE plots of BANKING77 test utterances when xml-RoBERTa is used to extract the embeddings.

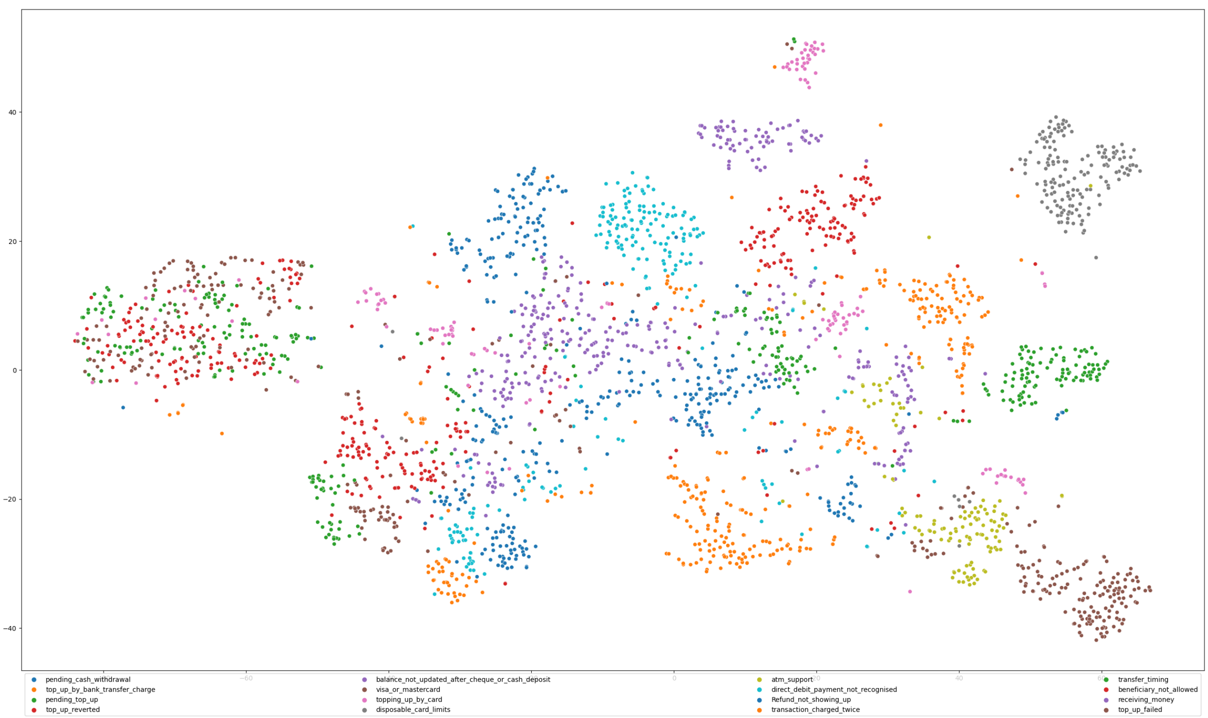

Figure 4: tSNE plots of BANKING77 test utterances when xml-RoBERTa - fine-tuned with the binary classification loss - is used to extract the embeddings.

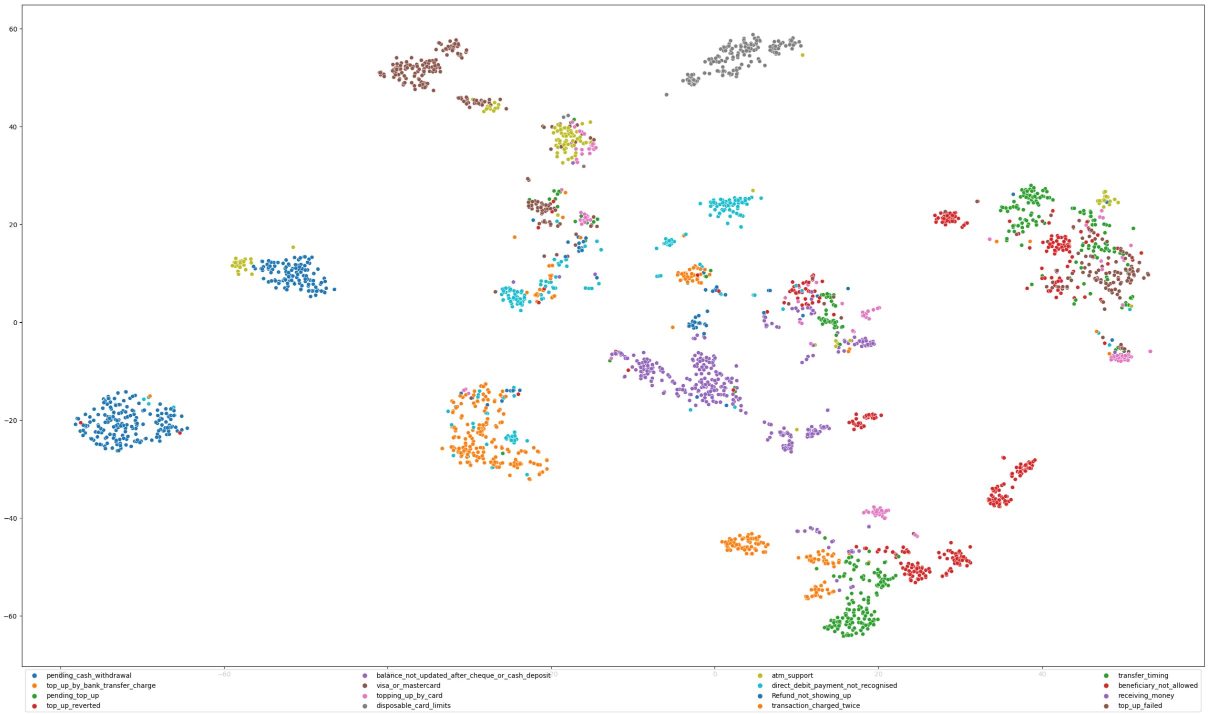

Figure 5: tSNE plots of BANKING77 test utterances when xml-RoBERTa - fine-tuned with the cosine similarity loss - is used to extract the embeddings.

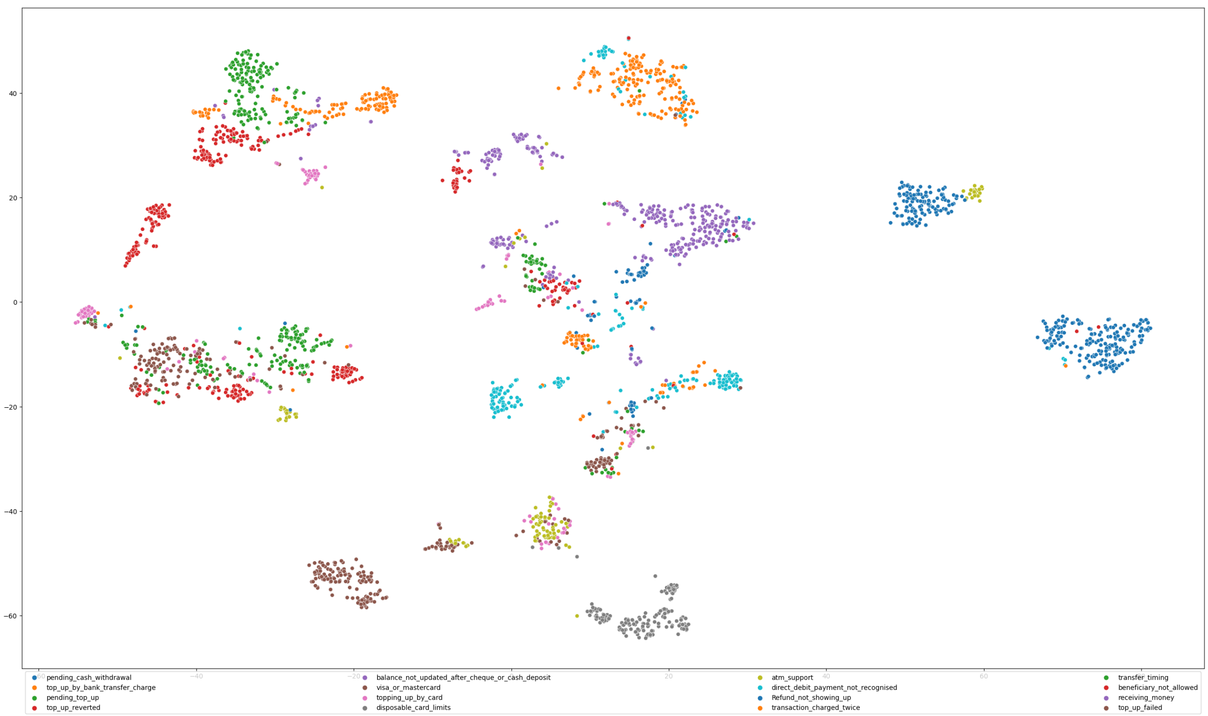

Figure 6: tSNE plots of BANKING77 test utterances when xml-RoBERTa - fine-tuned with the contrastive learning loss - is used to extract the embeddings.

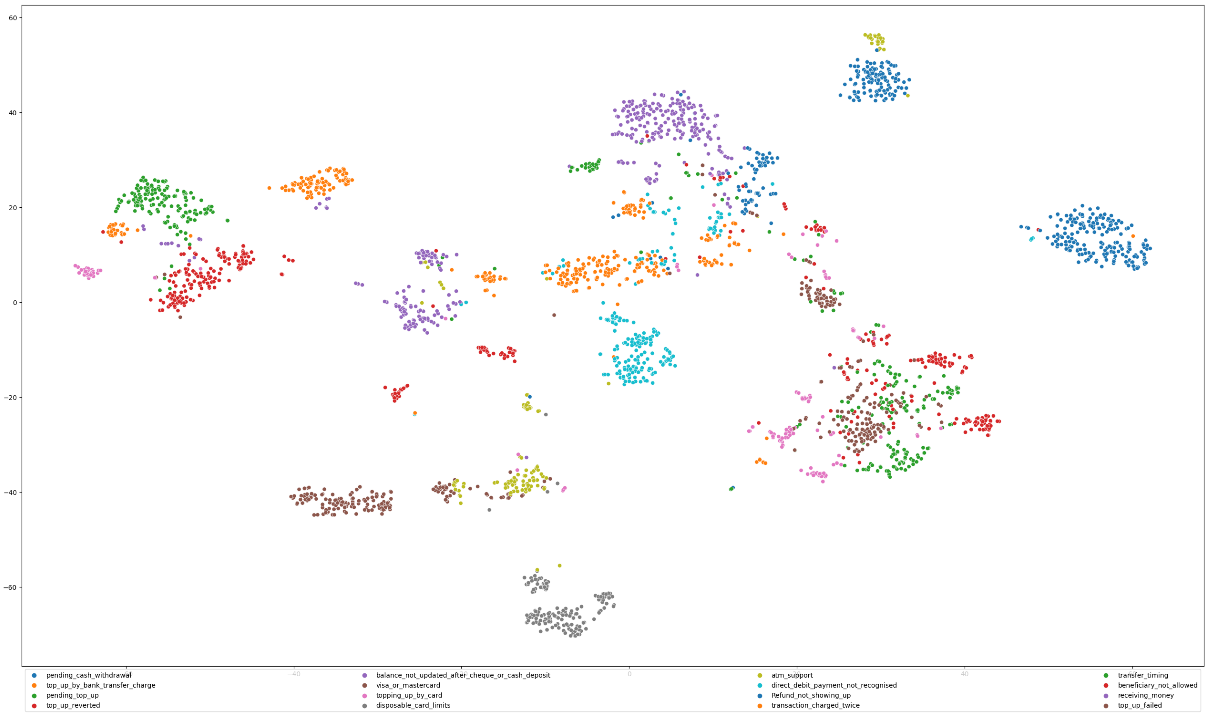

Figure 7: tSNE plots of BANKING77 test utterances when xml-RoBERTa - fine-tuned with the triplet margin loss - is used to extract the embeddings.

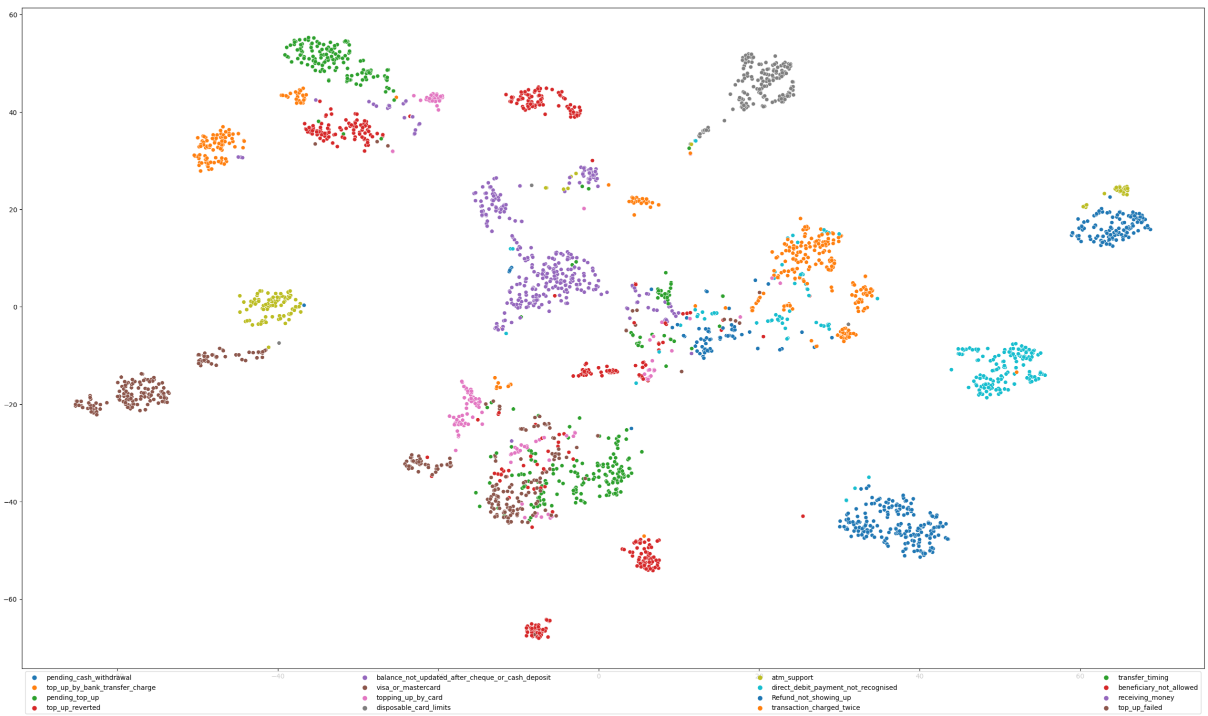

Figure 8: tSNE plots of BANKING77 test utterances when xml-RoBERTa - fine-tuned with the supervised clustering loss - is used to extract the embeddings.