Methods for Semi-Supervised Data Augmentation in Sequence Classification Tasks

Brian Hsu
Master of Business Analytics
Massachusetts Institute of Technology
email@domain

Elaine Chen
Master of Business Analytics
Massachusetts Institute of Technology
xtchen64@gmail.com

Abstract

Our project objective is designing a system for scalable and accurate unsupervised/semi-supervised label generation for text classification data. A common data issue across a variety of contexts and industries is the lack of well-labeled data and a prominent example is intent classification, where customers may send texts/emails about a variety of problems. With enough labeled intent data, it is entirely possible to train from scratch or safely leverage transfer learning. However, obtaining enough labeled data points for ground-up development or even fine-tuning can be challenging in terms of cost and manual effort. Instead, we aim to investigate generalizable methodologies for using few-shot and zero-shot for labeling at scale with limited labeled data.

1 Literature Review

Our project started by investigating zero-shot learning and few-shot learning methods on text sequence classification tasks. Although such tasks are generally hard due to lack of data, there are many extensively investigated models:

In zero-shot learning, models like facebook's BART-MNLI (Wenpeng Yin, 2019) do a good job in assigning probabilities to label classes based on one query only. In few-shot learning, Siamese Neural Network(SNN) (Florian Schroff, 2015) is popular within the computer vision community and generally predicts well in small dataset settings. Pseudo Siamese Neural Network(PSNN) (Xia et al., 2021) further allows for different weight optimizations for queries and intents, which is more suitable for our scope, since queries and intents are different types of texts.

To find a meaningful embedding space for the queries and intents, we explored different pretrained embeddings such as GloVe (Pennington et al., 2014), DistilBERT (Sanh et al., 2020) and Sentence-BERT(SBERT) (Reimers and Gurevych, 2019). Particularly, SBERT uses SNN during the pretraining phase to enhance the performance of BERT embeddings on sentence-related tasks. This gives us confidence that using a combination of SBERT embeddings and SNN/PSNN would recover a more powerful framework on query-intent classification tasks.

The scope and architecture of our project has changed a lot as we make progress in the modeling. Inspired by various literature and designed our own pipeline, in retrospect, our framework ended up resembling the few-Shot curriculum learning method proposed by Wei et al. (Wei et al., 2021). Specifically, we implemented a pipeline that is similar to the 2-stage model (without the gradual curriculum learning). This is an interesting coincidence and gives us confidence that our initial hinge that using reinforcement learning might enhance the result even more, although this is out of the scope of this project.

2 Data

We will be working with the Banking77 data (Casanueva et al., 2020), created by PolyAI and made available on HuggingFace. This data contains over 10,000 labeled queries with 77 unique intents related to the banking domain. Some examples are given below:

- Query: I tried to top up, but it didn't finish.
  Label: pending top up
- Query: I returned something to a store but can't see my refund.
  Label: refund not showing up
- Query: I am still waiting on my card?
  Label: card arrival

Since the dataset itself focuses on fine-grained single-domain intent detection, we expect the task to be specific to the finance and banking domain. This means that most of the out-of-pocket state-of-art models need to be fine-tuned and/or enhanced in order for us to achieve a better performance.

A typical problem that exists in the domain is the high cost and high difficulty of high-quality data labeling. Therefore, our project is well-positioned to tackle the small data problem with this dataset.

One specific difficulty in the domain is that the intents themselves are often semantically close to each other, which can be confusing to most of the models that leverages the semantic embeddings of the intents. There needs to be a high level of separation of queries that belong to different intents, as well as a separation between intents, in order for our model to successfully categorize the queries to the correct labels.

To reduce confusions to the models by the number of label categories, we have limited the data to 30 labels. We also designated three regimes for learning. Our unsupervised data is defined only as knowing the set of 30 labels (no queries or query-label pairs). Our semi-supervised and supervised data consists of about 20 labeled queries for each of the 30 intents (600 labeled observations)[1]. We also further shrank the dataset down to study the effect of data size and the sensitivity of our conclusions. This will be elaborated in the following sections.

3 Model Methodology

The scope of our project involves creating intent labeling mechanisms under the three data limitations/learning regimes mentioned above (unsupervised, semi-supervised, supervised). In total, we devised 10 modeling schemes, some of which build upon one another. The primary motivation is that some learning methodologies (zero-shot and few-shot) tend to work well under small-data regimes and we seek to understand their strengths and limitations under varying degrees of training data availability.

The unsupervised methods use pre-trained embeddings (GloVe, SBERT) to generate pooled representations of both the queries and the labels. Then we calculate similarity score (the inner products, for example) of query-label pairs, and take the pair with the highest score.

[1]We selected up to 20 examples because we feel this is reasonably achievable for any domain (e.g. an individual can reasonably come up with that many different ways to inquire about a topic)

Similar to unsupervised learning, the semi-supervised methods leverages pre-trained embeddings (e.g. from SBERT). Moreover, we trained on the small dataset to enhance the results. Examples of the techniques that we have tried includes Siamese neural networks (SNN), Pseudo Siamese neural networks (PSNN), zero-shot learning (with centroids or PSNN).

Moreover, we took one step further and generated an "augmented dataset" to enhance the accuracy of the best-performing model (Fine-tuned DistilBERT) even more. Our hypothesis is that an ensemble model that generates "pseudo-labels" from unlabeled data will help improve model performance when added into the training data, especially if we filter for the pseudo-labels with high confidence of correctness only.

Finally, we used DistilBERT model for supervised learning. We fine-tuned the model on the same training dataset and conducted the classification task. We show the results of all these models under different training data limitations (of 240 obs, 360 obs, and 600 obs) in Section 4.1.

For implementation, we use Google Colab Pro and have enough computation power/memory for all the intended experiments, thanks to the small size of the training datasets.

3.1 Unsupervised

3.1.1 GloVe Similarity

We used GloVe embeddings (Pennington et al., 2014) as the word representations for both queries and labels. From that, we calculated a similarity score based on inner products to identify the most probable label for each query. We then make the label prediction by choosing the pair with the highest similarity score.

3.1.2 SBERT Query-Label Similarity

Sentence-BERT (SBERT) is a modification of the pretrained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity (Reimers and Gurevych, 2019). We use the out-of-the-box SBERT embeddings and adopt a similar method as for GloVe to classify the label for each query.

3.2 Semi-supervised

There are two categories of training strategies we focus on - zero shot and few shot learning. In

zero-shot learning, models like BART-MNLI (Wenpeng Yin, 2019) are trained through entailment modeling. In few-shot learning, one of our highlights is extending the SBERT idea with Pseudo-Siamese neural network (PSNN).

3.2.1 Nearest Centroid

In the centroid method, we create centroids for each label by averaging the SBERT embeddings for all queries corresponding to that label (giving us 30 centroids). To make predictions on a new query, we embed it with SBERT and label it based on the nearest label centroid.

3.2.2 Siamese Neural Networks (SNN)

We designed a Siamese Neural Network inspired by the motivation behind Facenet (Florian Schroff, 2015). The main idea is that just as one person can be represented by pictures from different angles, one intent can be represented from different wordings. We built the SNN in PyTorch by taking the SBERT model and then further defining a feed-forward layer to tailor the embeddings for our purposes.

In the training process, we implemented the hard-negative and hard-positive mining process to generate online training batches. We also made a small modification by also training the network so that we can closely embed queries and their respective centroids. In other words, we have (anchor query, query(+), query(-)) triplets and also (anchor query, centroid(+), centroid(-)) triplets. When we make predictions, we feed a new query through the SNN and also feed in all centroids (made in the same manner as described in the previous section). Then, we compute the closest centroid in Euclidean norm and use its respective label as our prediction. To accomodate the centroids in training, we therefore also have a notion of hard-positives/negatives for query-centroid pairs.

3.2.3 Pseudo-Siamese Neural Network (PSNN)

The Pseudo-Siamese Neural Network (PSNN) structure involves only a slight modification on the SNN structure, but tends to work well. Because we need to embed both queries and centroids, we reasoned that using different feed-forward layers (one for queries, one for centroids) would make more sense. This is illustrated in the diagram in Figure 1.

When training the SNN and PSNN, we considered several things. Most important were the structure of the network (needed something that doesn't overfit the training data), the margin of separation (how far apart embeddings should be), and the optimizer (we chose Adam). When back-propagating the gradients, we froze the SBERT model that produced the initial embeddings because allowing it to fine-tune required greater GPU memory than what we had available.

We visually illustrate the effectiveness of the PSNN for creating well-separated embeddings for different labels in Figure 2. To illustrate the effect of PSNN in terms of how much enhancement it brings to the classification task, we look at how well-separated the query embeddings are just using SBERT (left) versus after feeding it through the trained PSNN (right). Particularly, we use T-SNE (van der Maaten and Hinton, 2008) as a dimension reduction method to visualize the distances of queries in 2-dimensional plots. In the plot, each point is a query and different colors correspond to different intents. As the figure illustrates, there is much better separation after the PSNN embeddings, showing that even with limited data, PSNN can effectively distinguish intents. It also demonstrates a difficulty in semi-supervised learning, as it has difficulty with separating intents that look similar such as pink and red (which represent the intent "balance not updating after bank transfer" and "balance not updating after cash/check deposit" respectively).

3.2.4 Zero-shot with top-3 from Centroid

This method was motivated by the idea that because ZSL models have never seen our specific data before, if we feed in the full set of hypotheses (i.e. all 30 labels), then it would likely get confused especially as some labels are semantically quite similar (e.g. refund not showing up vs. refund not correct). Hence, if we filter down the full set of hypotheses to a few (e.g. 3) relevant ones, we would expect a ZSL model to make better predictions. One way to get 3 of the most relevant intents is to use the 3 closest centroids from the nearest centroid method.

3.2.5 Zero-shot with top-3 from PSNN

Similar to the previous section, another way to obtain the top three relevant candidate intents is to use the PSNN model. Recall that we make predictions based on finding the nearest centroid PSNN-embeddings to the PSNN-embedded query. Similarly, we can gather the three closest centroids and

Figure 1: Model Architecture of PSNN

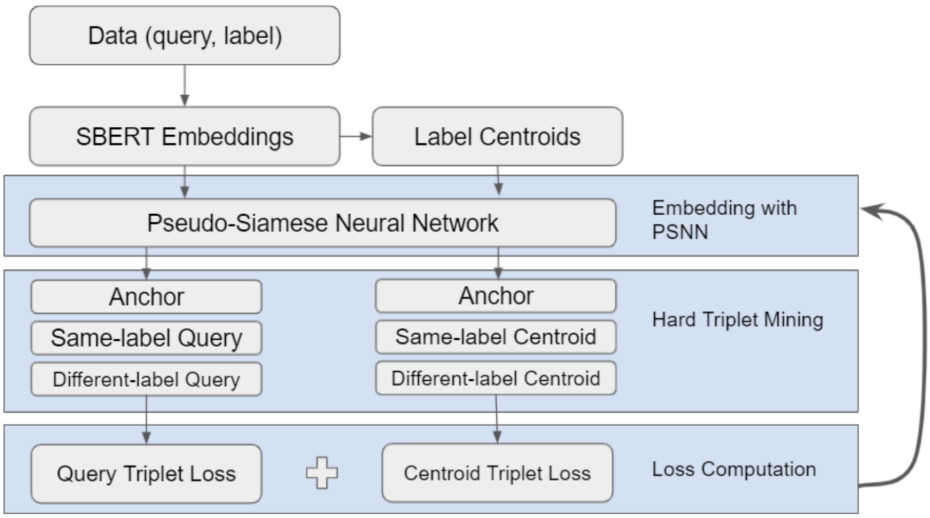
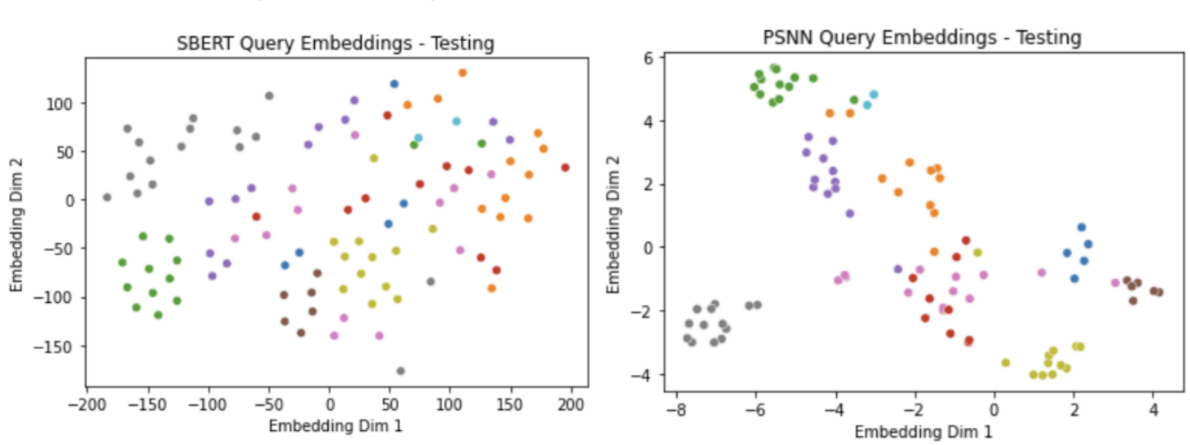


Figure 2: Dimension-reduced cluster separation before and after PSNN



use those labels as our hypotheses set.

3.2.6 Ensemble of semi-supervised

The semi-supervised models above has different criteria, and often different predictions and probability (or confidence). To create an ensemble, we looked at the three best performing models (SBERT nearest neighbors, Nearest Centroid, PSNN) and used a softmax function to produce probabilistic outputs for each label across all queries. In other words, each query would have 30 (one for each label) probabilistic predictions for each of these models. We then trained a random forest model on the limited training data (still 240, 360 or 600 observations) through a CV random grid-search. We chose to use a random forest here after experimenting with a couple different models, including multilayer perceptions and multinomial logistic regression. Importantly, we need to pick a ensemble model that does not overfit the data especially because we have so few training instances. Due to the relative ease of fitting and tuning random forests, we decided to use this ensemble method.

The advantage of an ensemble is two-fold. First, ensemble methods are a generally helpful method to improving predictions. Second, when used for predictions the ensemble method gives a corresponding probability score which can be interpreted as a degree of confidence. This notion of confidence is important for augmenting our limited training data as we only want to augment with predictions where we are reasonably sure that the pseudolabel (from the ensemble model) matches the ground truth label.

3.3 Supervised

3.3.1 Fine-tuned DistilBERT

DistilBERT (Sanh et al., 2020) is a smaller version of BERT, which is fine-tuned with our own dataset. We are treating DistilBERT as an out-of-box method for sequence classification that generates results relatively easily comparing to the previously described semi-supervised methods.

Although we started with the same small dataset for the DistilBERT model, we call it a "supervised" method because we will eventually run it with an augmented dataset with larger size, as described below.

3.3.2 Fine-tuned DistilBERT with enhanced dataset

We still used the DistilBERT model. However, this time we fine-tuned on the augmented dataset generated with predictions on holdout data from the ensemble model. The ensemble model predicts a label (we call this a pseudolabel, as it could be different from the ground-truth) and a probabily. To gather the most confident predictions, we rank all predictions for each label based on the confidence, and take the most confident qeuries as our augmentation data. We are interested to learn about the impact of this augmented dataset to the performance on the test set.

4 Results

4.1 Classification Accuracy

We implemented the unsupervised, semi-supervised and supervised models described in previous sessions. Firstly, we used a training dataset of 600 observations, which belong to 30 different labels. We report the results in two separate tables - one for the performance on the original training set, and one for the augmented training set (number of observations are shown inside each table).

Next, we reduced the already-small dataset from 600 observations down to 360 and 240 observations, re-ran the experiments to examine the effect of dataset size. The results for each dataset size are shown in the 6 tables below (note that we are reporting the out-of-sample classification accuracy).

4.2 Discussion

4.2.1 Results Overview

Based on results, a majority of the models have a relatively high accuracy. GloVe serves as a baseline with poor performance, which is somewhat expected. SBERT similarity has a much higher accuracy in comparison. The semi-supervised models are expected to have even higher accuracy. Originally, we hypothesized that feeding the likely candidates intents (e.g. the top 5 neighbors from SBERT similarity) into ZSL would allow ZSL to better guess the true intent. However, we unexpectedly observe that sometimes the pre-trained ZSL model can get confused by the candidate set and perform worse than if we had just used the top most likely candidate from SBERT instead. This could be due to factors such as the specific wording of the hypothesis, which we may experiment with.

4.2.2 Performance across learning regimes

Given the small training dataset, our initial hypothesis was that semi-supervised methods might work better than unsupervised and supervised ones. This is because it leverages both prior knowledge on English word contextual embeddings and the dataset-specific knowledge such as query-intent linkages, in a flexible way (as we are trying multiple different models such as PSNN).

In general, the results align to our expectations. The unsupervised models do not change in accuracy performance because they don't use the training data to begin with. The semisupervised methods show some sensitivity as we reduce the training data, ranging from 74-82%. Finally, the supervised method (DistilBERT fine tuning) shows very high sensitivity as we reduce the training data, ranging from 65-85%. We figure that the sensitivity of the supervised method primarily involves the idea that it is prone to overfitting the data, but without overfitting, it does not gain much information from training. Hence, this behavior highlights the need for data augmentation. Next, we discuss the ensemble model and how we use it to produce an augmented dataset that can help the supervised model.

4.2.3 Performance of ensemble model

It is worth pointing out that the performance of ensemble model is very similar to the best model (PSNN) within the ensemble. This is interesting because the ensemble technique would often improve the performance, given "the wisdom of crowd". This might suggest that the PSNN model is already flexible and smart enough to achieve a top performance of its kind, and ensembling could not help much.

However, the ensemble model is still helpful

Table 1: Performance of unsupervised & semi-supervised methods (600 obs., 30 labels)

Training Scheme	Model	Accuracy (600 obs, 30 labels)
Unsupervised	GloVe Similarity	2.6 %
	SBERT Query-Label Similarity	46.6 %
Semi-supervised	Nearest Centroid	54.0%
	Siamese Neural Network	81.5%
	Pseudo-Siamese Neural Network	82.6%
	Zero-shot with top-3 from Centroid	65.4%
	Zero-shot with top-3 from PSNN	71.2%
	Ensemble of semi-supervised	82.9%

Table 2: Performance of supervised methods (600 obs., 30 labels)

Training Scheme	Model	Accuracy
Supervised	DistilBERT (600 obs.)	84.8 %
	DistilBERT with Augmented Data (1852 obs)	85.7%

Table 3: Performance of unsupervised & semi-supervised methods (360 obs., 30 labels)

Training Scheme	Model	Accuracy (360 obs, 30 labels)
Unsupervised	GloVe Similarity	2.6 %
	SBERT Query-Label Similarity	46.6 %
Semi-supervised	Nearest Centroid	52.8%
	Siamese Neural Network	81.3%
	Pseudo-Siamese Neural Network	79.3%
	Zero-shot with top-3 from Centroid	63.7%
	Zero-shot with top-3 from PSNN	69.3%
	Ensemble of semi-supervised	79.3%

Table 4: Performance of supervised methods (360 obs., 30 labels)

Training Scheme	Model	Accuracy
Supervised	DistilBERT (360 obs.)	75.5 %
	DistilBERT with Augmented Data (1125 obs)	82.3 %

Table 5: Performance of unsupervised & semi-supervised methods (240 obs., 30 labels)

Training Scheme	Model	Accuracy (240 obs, 30 labels)
Unsupervised	GloVe Similarity	2.6 %
	SBERT Query-Label Similarity	46.6 %
Semi-supervised	Nearest Centroid	54.2%
	Siamese Neural Network	74.6%
	Pseudo-Siamese Neural Network	74.8%
	Zero-shot with top-3 from Centroid	61.6%
	Zero-shot with top-3 from PSNN	66.4%
	Ensemble of semi-supervised	74.7%

Table 6: Performance of supervised methods (240 obs., 30 labels)

Training Scheme	Model	Accuracy
Supervised	DistilBERT (240 obs.)	66.5 %
	DistilBERT with Augmented Data (546 obs)	74.8 %

as we would use it to generate high-confidence pseudo-labels on unlabeled queries. We discuss the results of the data augmentation for the supervised models.

4.2.4 Discussion on augmented dataset

Using an augmented dataset on DistilBERT model does improve the result, regardless of the size of training data that we start with. However, the magnitude of improvement depends highly on how much data we begin with. The key finding of the project is that the improvement is more significant as the size of the training dataset decreases. The result confirms our expectation that data augmentation is more helpful when the lack of labeled data is more severe, which is exactly the situation we aimed to model.

Another consideration here is how exactly one defines a "confident prediction" from the ensemble model. In our case, we took a conservative approach and used the predictions with the highest probabilities for each label (the ensemble output a probability of exactly 1.0 for these confident predictions). Intuitively, as one lowers the tolerance of label distortions, one would expect the augmented data to get noisier and noisier, which would do the supervised model more harm than good. We consider it future work to experiment with different ways to define confidence. Some ideas are to use some prior notion of confidence (e.g. calibrate a probability threshold from the training data) or to use reinforcement learning (more in Section 4.3

4.2.5 Effect of training set size

As expected, the accuracy of model predictions drop as we shrink the dataset size. Specifically, the best-performing model (DistilBERT with augmented dataset) has an out-of-sample accuracy of 86% on the 600-observation training set and only 74.8% on the 240-observation training set.

However, it's worth pointing out that even with 240 observations our model still perform pretty well (74.8% accuracy). This shows that our framework of semi-supervised data augmentation succeeds in sequence classification tasks even in very small dataset scenarios.

4.3 Future Work

As a summary, our work suggests a pipeline and framework that leverages unsupervised/semi-supervised/supervised techniques to conduct sequence classifications when there is few or no data. The result can be generalized to other tasks of similar nature and applied in various industries where high-quality data labeling is expensive or difficult.

While we have explored many unsupervised, semi-supervised and supervised models, one interesting direction for future work is to try out Re-

inforcement Learning techniques. By setting up a proper reward mechanism, the agent would learn a strategy that classifies the queries to respective labels. Although we have not implemented it, we expect this approach to be computationally expensive, since the model needs to be retrained every time the agent makes a choice.

Another promising direction is to explore different ensemble methods when generating the augmented dataset, as well as when fitting the final model. For example, we have already experimented with ways of selecting the "most confident" pseudo-labels; however, we imagine that this process could be further refined.

References

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

James Philbin Florian Schroff, Dmitry Kalenichenko. 2015. Facenet: A unified embedding for face recognition and clustering.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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*. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Jason Wei, Chengyu Huang, Soroush Vosoughi, Yu Cheng, and Shiqi Xu. 2021. Few-shot text classification with triplet networks, data augmentation, and curriculum learning.

Dan Roth Wenpeng Yin, Jamaal Hay. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach.

Congying Xia, Caiming Xiong, and Philip Yu. 2021. Pseudo siamese network for few-shot intent generation.