Searching Near and Far for Examples in Data Augmentation

Anonymous EMNLP submission

Abstract

In this work, we demonstrate that augmenting a dataset with examples that are far from the initial training set can lead to significant improvements in test set accuracy. We draw on the similarity of deep neural networks and nearest neighbor models. Like a nearest neighbor classifier, we show that, for any test example, augmentation with a single, nearby training example of the same label–followed by retraining–is often sufficient for a BERT-based model to correctly classify the test example. In light of this result, we devise FRANN, an algorithm that attempts to cover the embedding space defined by the trained model with training examples. Empirically, we show that FRANN, and its variant FRANNK, construct augmented datasets that lead to models with higher test set accuracy than either uncertainty sampling or a random augmentation baseline.

1 Introduction

Despite super-human performance on benchmark datasets, state-of-the-art natural language processing models are far from true language understanding. Their brittleness has been demonstrated in many ways: simple rules can be utilized to create examples that cause trained models to fail, and methods that exploit model confidence can be used to generate nonsensical adversaries (Ribeiro et al., 2018; Alzantot et al., 2018; Jia and Liang, 2017). Modern techniques, coupled with manual effort, have even been used to generate examples on which production models fail (Ribeiro et al., 2020).

These issues are even more pronounced in cases when training data is scarce. Small training sets are common when developing domain-specific models, e.g., when building business-grade conversational systems (Coucke et al., 2018). In these cases, developers must construct their own training sets, which is costly and may introduce undesirable artifacts.

A family of approaches for combating brittleness, especially in scarce-data regimes, is data augmentation. A data augmentation algorithm is a mechanism for adding additional examples to the training set. The hope is that a model trained on the augmented data will be less prone to failure than a model trained on the original set. Algorithms for data augmentation in NLP have enjoyed success, but they are often specific to particular types of model failures and may require significant manual effort (Min et al., 2020; Li et al., 2020; McCoy et al., 2019; Kaushik et al., 2020).

Our goal is to develop a characterization of the examples, which, upon augmentation, are likely to improve test set accuracy. Drawing on the similarity between deep neural models and nearest neighbor models (Cohen et al., 2019), we study a BERT-based classifier, the examples on which it fails, and the nearest neighbors of those failures in a held-out set of examples. Similar to a 1-nearest neighbor classifier, we show that in 70% of experiments, augmenting a training set with a *single* nearest neighbor of a failed test example, leads to correct classification of the failure after re-training.

Bolstered by this result, we introduce FRANN, a data augmentation policy that attempts to "cover" the relevant regions of embedding space with training examples, so that nearest neighbor classification is effective. Specifically, FRANN operates by iteratively augmenting a training set with the most different example—measured by Euclidean distance in the model's embedding space—from the existing training examples. We compare FRANN, and its variant, FRANNK, to uncertainty sampling (active learning) and random augmentation from a held-out set (Settles, 2012). Our experiments show that augmentation with far-away examples leads to larger gains in test set accuracy than the competing methods.

2 Background

In this work, we study methods for data augmentation. In particular, we are concerned with scenarios in which the amount of labeled data is small. To ground our study, we focus on intent classification because it is a key step in building domain-specific conversational agents (Coucke et al., 2018). In practice, building such models are plagued by having very little training data[1]. At a high-level, in intent classification, the input is a natural language clause—called an *utterance*—and the goal is to predict its label.

We experiment with two datasets: Banking77 (Casanueva et al., 2020a) and CLINC (Larson et al., 2019). Banking77 includes 10,003 training utterances and 3080 test utterances unevenly distributed among 77 classes. CLINC includes 1500 `in scope` training utterances and 4500 test utterances evenly distributed among 150 classes. Like previous work, we ignore CLINC's `out of scope` utterances (Lee et al., 2021). For both datasets, we follow previous work and downsample the training data to a maximum of 10 utterances per class to mimic real-world intent classification settings (Anaby-Tavor et al., 2020; Larson et al., 2019; Casanueva et al., 2020b). The excluded training utterances are referred to as the *held out train set*, and we use them for augmentation.

3 Experiments

Recall that our goal is to characterize the examples which, when added to a dataset, yield the largest improvements in test set accuracy. In developing this characterization, we are inspired by the similarity between deep neural classifiers and latent space 1-nearest neighbor classifiers. In this section, we examine this similarity in the context of data augmentation.We begin by defining notation.

Notation. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be a dataset of pairs of points, $x \in X$, and their labels $y \in \{1, \cdots, K\}$, and let $f : X \to \{1, \cdots, K\}$ be a classification model. In our experiments we distinguish between train, test, and a held-out dataset using subscripts, e.g., $\mathcal{D}_{\text{train}}$. For some dataset $\mathcal{D}$, let $\mathcal{D}^+$ and $\mathcal{D}^-$ be the set of point-label pairs that are classified correctly and incorrectly by the model, respectively, i.e., $D^+ = \{(x, y) \in \mathcal{D} | f(x) = y\}$ and where y is the ground-truth label for the point x. Finally, let $\mathcal{D}[y'] = \{(x, y) \in \mathcal{D} | y = y'\}$, i.e., all examples in $\mathcal{D}$ with label y'. Throughout our experiments we represent each example, x, as its encoding in a trained model's final, pre-softmax layer. We measure distance between the embedded examples using Euclidean distance.

[1]A handful of industry practitioners building such system confirm this claim.

3.1 Augmentation with a single example.

Consider a 1-nearest neighbor classifier and a misclassified point-label pair, $(x, y) \in \mathcal{D}^-_{\text{test}}$. In order to correctly classify x, a new data point-label pair, $(x^\star, y)$ must be added to the training set such that,

$$(x^\star, y) = \operatorname*{arg\,min}_{(x', y') \in \mathcal{D}'_{\text{train}}} d(x', x)$$

where $d(\cdot, \cdot)$ represents Euclidean distance, and $\mathcal{D}'_{\text{train}}$ is the original training set augmented with a single example, $(x^\star, y)$. In words, $x^\star$ must be closer to x than any other point in the training set, and it must have label y.

We hypothesize that deep neural networks exhibit similar behavior with respect to data augmentation. Namely, that augmenting a dataset with a misclassified point's nearest neighbor (with respect to the model's embedding space), and training a new model on the augmented dataset, will yield a corrected prediction. Note that after augmentation, the training set only has one additional point.

To test this hypothesis, we fine-tune a (HuggingFace) BERT-base-uncased model (Devlin et al., 2018; Wolf et al., 2020) with an additional sequence classification layer on the (downsampled) CLINC training data (Section 2).We use the trained model to predict the labels of points in the test set, $\mathcal{D}_{\text{test}}$. For each misclassifed point-label pair $(x, y) \in \mathcal{D}^-_{\text{test}}$, we search for the nearest neighbor of x among the points in the held out training set of class y, i.e.,

$$(z^\star, y) = \operatorname*{arg\,min}_{(z, y) \in \mathcal{D}_{\text{heldout}}[y]} d(z, x).$$

If $z^\star$ is closer to x than *any* point in the training set, we create a new dataset, $\mathcal{D}'_{\text{train}} = \mathcal{D}_{\text{train}} \cup \{(z^\star, y)\}$. We refer to this method of selecting examples for augmentation as KNN. Moreover, if such an augmentation can be made, we train a (new) model on the augmented dataset and check whether the new model correctly classifies x, i.e., $f'(x) = y$. We repeat this process for all test points incorrectly classified by the initial model.

Result. Despite being trained on only 10% of the training examples, the initial fine-tuned model achieves 89% in-scope accuracy. The process described above (KNN) yields 297 augmented datasets. In 212 out of 297 experiments (71.4%), adding the nearest neighbor of the incorrectly classified test point, x, and re-training, yields a new model that correctly classifies x. This result demonstrates the potential impact of a single augmented example (especially in low-data regimes) and provides evidence of the similarity between our original model and a 1-nearest neighbor classifier in its learned latent space, *even* after re-training.

3.2 Augmenting with Multiple Examples

In our next experiment, we test whether the same phenomenon holds as the number of examples added to the training set grows. Since training alters the representations of all examples, this may break many of the nearest-neighbor relationships that exist before augmentation and re-training.

We conduct the following experiment with CLINC. We select a batch of misclassified examples from the test set, and for each example, we add a single held-out example to the training set using KNN (as above, Section 3.1). Thus, the number of examples added to the train set is exactly equal to the number of examples in the batch. The selected examples from the held-out set are all added to the train set simultaneously. Afterward, we train a new model on the augmented train set and calculate the fraction of test points from the batch that are correctly classified by the new model. We compare the examples selected by KNN with a policy that, for each test example of label y in the batch, selects a single example uniformly at random from $\mathcal{D}_{\text{heldout}}[y]$, i.e., the examples in the held-out set of label y. The result is visualized for batches of size $\{10, 30, 50, \cdots, 290\}$ in Figure 1. The chart shows that augmentation via KNN yields models that correctly classify $\sim$80% of previously misclassified target examples, regardless of the number examples added to the training set. This is consistently $\sim$2x better than selecting random examples of the same classes as the target test examples. We observe a similar trend when this experiment is repeated on the Banking data (Figure 3, Appendix).

3.3 Augmentation Policies

The experiments above show that augmenting a training set with the nearest neighbor of a failing test example (i.e., KNN) often leads to correct classification of that test example after retraining. However, KNN requires: 1. knowledge of failing test points, and 2. held-out, *labeled* data, neither of which is likely to be available. On the other hand, recent work shows that augmentation via retrieval

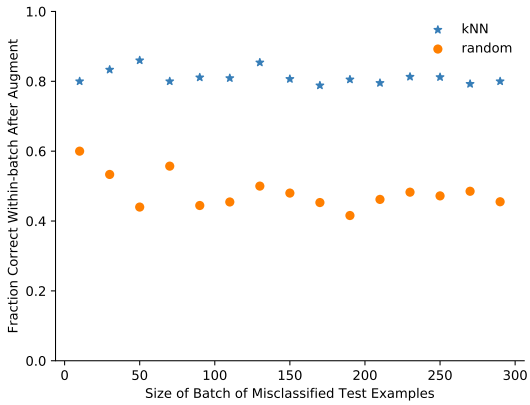

Figure 1: **Fraction Correct - CLINC.** The fraction of incorrectly predicted test examples that are predicted correctly after augmentation (KNN) and re-training.

from an *unlabeled* corpus can be effective for improving test accuracy (Du et al., 2020). Thus, in our final experiment, we study a variation of KNN for settings in which an unlabeled corpus is available.

We propose FRANN, The FaRthest Nearest Neighbor algorithm, that attempts to "cover" the latent space with examples. Intuitively, by covering the space, it is more likely for each test point to have a nearby neighbor in the training set. By the experiments above, this is likely to increase test set accuracy. To cover the space, FRANN selects unlabeled examples, greedily, in *decreasing* order of distance to their nearest neighbor in the training set. We also test two variants: FRANNK and FRAALL, which greedily select unlabeled examples in descending order of average distance to their closest k neighbors, and to all training examples, respectively. We compare our algorithms to an uncertainty sampling (ENTROPY), i.e., greedily selecting unlabeled examples in descending order of entropy in the trained model's corresponding softmax distribution (Settles, 2012). As a baseline, we also consider an algorithm that randomly selects unlabeled examples (RANDOM). In practice, the augmented unlabeled examples can be automatically labeled by a separate model (Du et al., 2020) or by hand. For simplicity, we use the ground-truth labels. We report test set accuracy as the number of augmented examples increases.

Figure 2 visualizes the result for the Banking dataset. The plot shows that FRANN and FRANNK are top performers, achieving the highest accuracy when 500 examples are used for augmentation. The gap between FRANN and

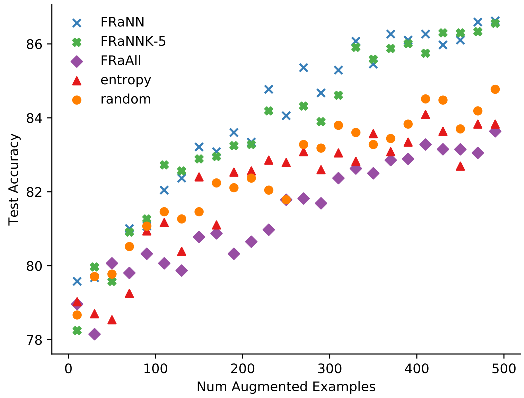

Figure 2: **Test Set Accuracy - Banking.** Test set accuracy as a function of the number of augmented examples for Banking dataset.

FRANNK and the rest of the policies increases with batch size. Interestingly, FRAALL performs worst for many batch sizes. Similar results for CLINC appear in the appendix (Figure 4).

4 Discussion

Our experiments underscore the value of augmenting a dataset with points that are *far* from the existing training examples. This is opposite of recent approaches, which augment a dataset with examples that are similar to the training set (Anaby-Tavor et al., 2020; Du et al., 2020). Understandably, augmenting with similar examples is safe; i.e., nearby examples are more likely to be in-domain and relevant. However, our work suggests that such a conservative approach is likely excluding examples that could significantly improve accuracy. Therefore, we conjecture that augmenting a dataset with both nearby and far-away points is likely to yield the largest improvements in test set accuracy.

Limitations. We raise a handful of limitations of our results. First, we only test a single model (fine-tuned BERT-base uncased) on a single task (intent classification). Given the similarities between neural and nearest neighbor models, we are optimistic about similar results holding for other tasks. Next, to mimic real-world scenarios, the training sets we use are small. Improvements from augmentation are likely more modest for larger training sets. Finally, we note that all of the examples we use for augmentation are (approximately) drawn from the test distribution. In practice, this would not be the case for a large unlabeled corpus. Despite this, we argue that our experiments are interesting in their own right, and demonstrate the value of far-away examples in data augmentation.

5 Related Work

Some studies of data augmentation in NLP introduce syntactic and semantic perturbations of training examples, which when used during augmentation, improves model robustness (Min et al., 2020; Li et al., 2020; McCoy et al., 2019). Related work demonstrates that augmenting a training set with counterfactual examples improves classifier performance, especially on counterfactual test examples (Kaushik et al., 2020). Neural language models have also been used to create new training examples by replacing tokens in original training instances (Kobayashi, 2018). Unlike these works, our method of augmentation specifically considers the model's encoding of the training set.

One closely related exploration studies nearest neighbors of misclassified test examples with respect to the train set (Rajani et al., 2020). Unlike our study, they focus on analyzing model predictions and finding labeling errors. They test the effect of excluding groups nearest neighbors from training, while we focus on augmentation. Moreover, we experiment with a handful of augmentation policies. A similar work was carried out from the lens of active learning lens by Geifman and El-Yaniv (2017). However, their work was limited to the field of computer vision.

Many recent studies demonstrate the effectiveness of utilizing nearest neighbors for various neural prediction tasks. For example, in sequence labeling, the nearest neighbors of a test sequence can be leveraged to accurately label the sequence (Wiseman and Stratos, 2019). A similar phenomenon was demonstrated in language modeling (Khandelwal et al., 2020). Like our work, in both of these cases, nearest neighbors are computed using distance in the learned latent space of a language model. However, both of these works focus on test time prediction using nearest neighbors rather than data augmentation. Other work with similar flavor includes neural machine translation, language generation, and text classification approaches that explicitly retrieve training examples at test time (Gu et al., 2018; Zhang et al., 2018; Weston et al., 2018; Wallace et al., 2018). Related work studies influence functions and their role in interpretability in NLP (Koh and Liang, 2017; Han et al., 2020).

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! *AAAI*, 34(05):7383–7390.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020a. 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.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020b. 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.

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2019. DNN or k-NN: That is the generalize vs. memorize question. *arXiv:1805.06822*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Ves Stoyanov, and Alexis Conneau. 2020. Self-training improves pre-training for natural language understanding. *CoRR*, abs/2010.02194.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Empirical Methods in Natural Language Processing*.

Yonatan Geifman and Ran El-Yaniv. 2017. Deep active learning over the long tail. *CoRR*, abs/1711.00941.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894, International Convention Centre, Sydney, Australia. PMLR.

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.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020.

Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352, Online. Association for Computational Linguistics.

Nazneen Fatema Rajani, Ben Krause, Wengpeng Yin, Tong Niu, Richard Socher, and Caiming Xiong. 2020. Explaining and improving model behavior with k nearest neighbor representations. *arXiv:2010.09030*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Burr Settles. 2012. Active learning. *Synthesis lectures on artificial intelligence and machine learning*, 6(1):1–114.

Eric Wallace, Shi Feng, and Jordan Boyd-Graber. 2018. Interpreting neural networks with nearest neighbors. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 136–144.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

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

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. 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 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.

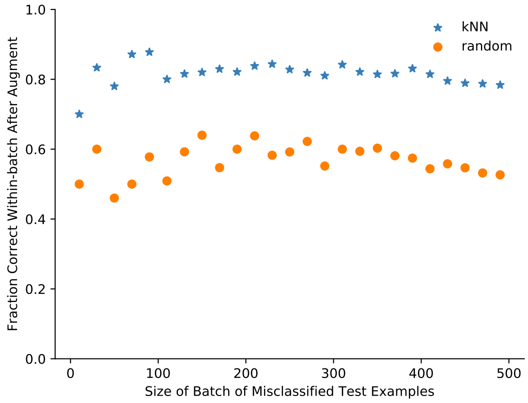

Figure 3: **Fraction Correct - Banking.** The fraction of incorrectly predicted test examples that are predicted correctly after augmentation (KNN) and re-training for Banking dataset.

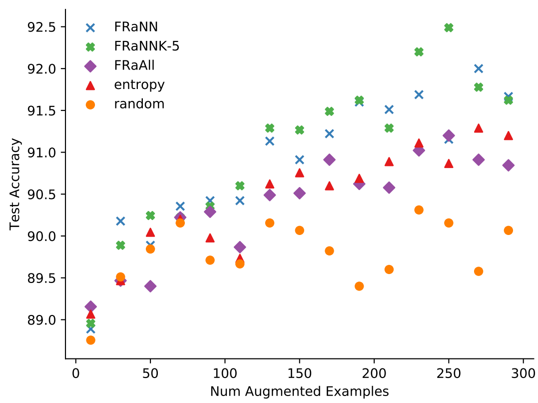

Figure 4: **Test Set Accuracy - CLINC.** The test set accuracy as a function of the number of augmentation points for data augmentation policies.

Appendix

A Augmenting with Multiple Examples

We follow the same methodology as presented in Section 3.2—augmenting with multiple examples in increasing batch sizes—but experiment with the Banking dataset. The result is visualized for batches of size $\{10, 30, 50, \cdots, 490\}$ in Figure 3.

Similar to on CLINC, the chart shows that augmentation via KNN leads to models that correctly classify $\sim$80% of previously misclassified target examples, regardless of the number examples added to the training set. This is consistently better than selecting random examples of the same classes as the target test examples.

B Other Augmentation Policies

In Section 3.3, we proposed 3 augmentation policies—FRANN, FRANNK and FRAALL—and compared them with uncertainty sampling (ENTROPY) and a RANDOM baseline. We perform the same for the CLINC dataset and visualize the result in Figure 4. The plot shows that FRANNK achieves the highest maximum held-out test accuracy (when 250 zpoints are augmented to the training set). After all 290 augmentations are made FRANNK and FRANN achieve similarly high accuracy, followed closely by ENTROPY. We hypothesize that our approach outperforms ENTROPY because deep-neural networks are notorious for having uncalibrated confidences (Guo et al., 2017; Feng et al., 2018). All policies outperform RANDOM augmentation. Together, the results reinforce the similarity between our BERT-based sequence classifier and a 1-nearest neighbor classier with respect to data augmentation, and suggest that augmentation with examples that are far away from the training examples helps improve test set accuracy more than the other methods.