A Flexible Clustering Pipeline for Mining Text Intentions

Xinyu Chen
Xinyu.Chen@verint.com
Verint Systems Inc.
Melville, NY, USA

Ian Beaver
Ian.Beaver@verint.com
Verint Systems Inc.
Melville, NY, USA

ABSTRACT
Mining the latent intentions from large volumes of natural language inputs is a key step to help data analysts design and refine Intelligent Virtual Assistants (IVAs) for customer service and sales support. We created a flexible and scalable clustering pipeline within the Verint Intent Manager (VIM) that integrates the fine-tuning of language models, a high performing k-NN library and community detection techniques to help analysts quickly surface and organize relevant user intentions from conversational texts. The fine-tuning step is necessary because pre-trained language models cannot encode texts to efficiently surface particular clustering structures when the target texts are from an unseen domain or the clustering task is not topic detection. We describe the pipeline and demonstrate its performance using BERT on three real-world text mining tasks. As deployed in the VIM application, this clustering pipeline produces high quality results, improving the performance of data analysts and reducing the time it takes to surface intentions from customer service data, thereby reducing the time it takes to build and deploy IVAs in new domains.

CCS CONCEPTS
• Computing methodologies → Information extraction; Cluster analysis.

KEYWORDS
fine-tuning, deep clustering, intent mining, community detection

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WWW ’22, April 25-29, 2022, Lyon, France
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . $15.00
https://doi.org/10.1145/nmmnm.nmmnm

1 INTRODUCTION
Intelligent Virtual Assistants (IVAs) are becoming more popular in customer service and product support tasks [10]. At Verint, the process to design and refine IVAs relies on data analysts who are familiar with specific terminology in a given language domain such as transportation or finance to mine customer service texts for latent user intentions, or intents. An intent is the interpretation of user input that allows the IVA to formulate the ‘best’ possible response and its detection is typically treated as a supervised classification problem in commercial IVAs [1]. It is common for these analysts to receive a large batch of customer service logs from a company who would like to deploy an IVA to help automate some aspect of their customer service. The analysts then need to mine this text data to surface the most common intents and determine which use cases an IVA would be able to easily automate. They would
then recommend a subset of high value intents for a company-specific IVA implementation to be deployed on a website, mobile application, or phone support line.

Verint Intent Manager (VIM) is a powerful language classification tool that helps our data analysts to review and organize such large volumes of unlabeled conversational text inputs into various intents. Figure 1 shows the main VIM work space where analysts search, filter, and group unlabeled text inputs (right table) and assign them to labels, then organize the labels into hierarchies (left sidebar)\(^1\). These labeled text inputs are then exported to build intent classifier models for production IVAs. As a manual drag and drop process is slow, using text clustering algorithms to provide recommendations for analysts to gain a better understanding of the type of language contained in input texts can partially automate the intent categorization process. We apply language models such as BERT, RoBERTa, and ELMo [4, 6, 8] to encode texts to contextualized word representations for clustering in VIM. However, there are three challenges that affect the performance of text clustering tasks on real-world customer service datasets.

First, the desired grouping of texts for describing user intentions may be different from those learnt by fully unsupervised approaches [11]. Unless given further guidance, the contextualized text representations from pre-trained language models are more suited for finding latent topic clusters than surfaceing user intentions [3]. Second, the size of unlabeled conversational logs obtained from contact centers varies from tens of thousands to millions. The performance of the clustering method needs easily scale to such large volumes so that analysts can have the clustering results fast enough to iterate between labeling and evaluation. Third, the true number of clusters on a new dataset is often unknown and analysts may get sub-optimal results when using an inaccurate number of clusters. Analysts then waste valuable time sorting through poor clustering results which defeats the purpose of automation.

To address these challenges for data exploration in the VIM platform, we developed an end-to-end text clustering pipeline that integrates four highly successful AI/ML techniques: transfer learning with state-of-the-art language models (e.g., BERT), a high performance library for k-nearest neighbor (k-NN) graph construction and k-means clustering leveraging multiple GPUs, and a fast community detection algorithm to uncover high quality community structures. In this paper we discuss how we overcame these three challenges in our production application using this pipeline.

2 THE PIPELINE

Figure 2 shows our pipeline for flexible text clustering\(^2\). The first component is the fine-tuning/embedding module. In the initial state when no data has been labelled yet, the untuned language model representations are directly used for downstream clustering as a fully unsupervised approach. As the analyst works within the VIM interface (see Figure 1) they begin to organize and label some of the texts from the unsupervised clustering result or from manually created regex patterns. Once the analyst has provided some small subset of labeled samples it is possible to fine-tune the language model to their particular task. To fine-tune a language model, we add a linear classifier on top of the mean pooling layer. We optimise the cross entropy loss function to update the language model’s parameters and the classifier at the same time. After this fine-tuning stage, we discard the linear classifier and use the updated language model and mean pooling to produce text representations.

If the user does not specify the desired number of clusters, the representations from the language model are used to generate a k-NN graph using the faiss library [5]. The faiss library can efficiently provide similarity search and clustering with GPU support. Given the maximum observed volume of input texts of VIM is in the millions of sentences, we can construct the k-NN graph by using exact searching with L2 distances in a few seconds.

We use the faiss k-means [5] or Louvain [2] depending on if the user specifies the desired number of clusters. The Louvain algorithm can detect hierarchical clustering structures from large networks with millions of nodes and billions of links. Both algorithms can quickly handle the volumes of text commonly worked on by VIM users. The clustering results are presented to analysts (see screenshots in Figure 4) where they can click on a cluster to see member texts and bulk assign clusters intention labels, manually edit cluster membership, or launch a sub-clustering task to create intent hierarchies. As the analyst iterates over labelling and clustering the language model can be additionally fine-tuned creating a system that adapts to how each analyst is organizing their dataset.

\(^1\)A video walk through of an older beta version of VIM, then named Prompt AI, given by a data analyst is available here: https://youtu.be/UjLu1L-ES0w

\(^2\)Source code for the pipeline and additional experiments are made available here: https://nextit-public.s3-us-west-2.amazonaws.com/FinetuneLM-Clustering.zip
A Flexible Clustering Pipeline for Mining Text Intentions
WWW ’22, April 25-29, 2022, Lyon, France

| Dataset  | Size  | # Clusters | Purity    | NMI      |
|----------|-------|------------|-----------|----------|
| RCV1     | 674k  | 20.6 (4.0)| 0.913 (0.944) | 0.672 (0.808) |
| STKOVFL  | 90k   | 35.4 (10.0)| 0.926 (0.926) | 0.737 (0.823) |
| RCT      | 270k  | 10.4 (5.0)| 0.787 (0.830) | 0.524 (0.608) |

Table 1: Clustering quality of Louvain and (k-means).

3 ANALYSIS AND DISCUSSION
In this section, we show the clustering results and try to answer the following questions regarding to our pipeline design: (1) How accurately can the pipeline separate different groups of texts? (2) Can Louvain detect high quality clusters without knowing the true number of groups? (3) How fast can the pipeline complete end-to-end clustering tasks on different sized datasets?

3.1 Clustering Accuracy
Clustering Purity and Normalized Mutual Information (NMI) are commonly used metrics for cluster quality evaluation [7], both range from 0.0 to 1.0. Higher purity means most texts belong to one true class in each detected cluster. NMI penalizes splitting data into a large number of small clusters to correct this shortcoming of the purity score. Higher NMI means good correlation between predicted cluster labels and true classes. In Table 1 we show an excerpt of three out of six real-world text datasets from an extensive evaluation of the pipeline we published in a previous work [3]. These numbers were achieved by fine-tuning BERT with 2.5% labeled samples of each dataset. In our additional experiments the pipeline achieved high clustering quality with both Louvain and k-means, and BERT and RoBERTa performed similarly but both outperformed ELMo.

3.2 Deep Analysis of Louvain
In Table 1 we showed the number of clusters detected by Louvain is usually greater than the true number of clusters, shown in parentheses. However, the high purity scores indicates the majority of documents within the same predicted clusters are from the same true classes. Because documents can have hierarchical topics, Louvain appears to be detecting more fine-grained cluster structures inside each high level group from the k-NN graph.

In Table 2, the top-5 frequent bigrams from the 9 clusters detected by Louvain are compared with the bigrams from the true cluster corresponding to the topic category of GCAT (government and social) from the RCV1 (Reuters News) dataset. One example is 736 news articles forms a separate group about Romania; 1,839 articles forms another group about weather. These two clusters consist of only 0.4% and 1.1% of the articles in the entire GCAT category respectively. Another example is the separation of news related to two world cup events. By seeing the surfaced bigrams, analysts can understand the first group is more likely about tennis games and the second group is about soccer games. The two takeaways are: the k-NN graph accurately connects articles with their nearest neighbors; second, Louvain accurately detects communities with high resolution. This means VIM users get good clustering performance even if they do not know or specify the number of clusters for their task.

3.3 Computation Time
We show the pipeline is horizontally and vertically scalable on more computation resources in Figure 3. The top graph shows the running time of the pipeline by stage on 673,000 Reuters News articles across different Amazon EC2 instance types. Using a p3.16xlarge instance (eight V100 GPUs), the end-to-end clustering running time is 5.9x faster than on the p3.2xlarge instance (one V100 GPU).

The bottom graph in Figure 3 shows the vertical scalability of the clustering pipeline. Of the three tasks, STKOVFL has over 90,000 sentences, which would be near the average dataset size for the users of VIM that we have observed in production so far. The RCT task resembles a larger dataset size analysts encounter quite often. The RCV1 task contains 0.67 million news articles, which is analogous to a large work load in VIM. When the pipeline is deployed on Amazon p3.16xlarge instances, the end-to-end computation can be finished under 5 minutes for the average tasks, taking closer to 15 minutes for the largest dataset.

4 CASE STUDY
In this section, we use a case study to show our pipeline provides similar insight to analysts unfamiliar with a dataset as can be discovered by an analyst who is. We demonstrate the clustering pipeline has improved the usability of the VIM application even without a
Top-5 Frequent Bigrams for the RCV1-GCAT Cluster by Clustering Method

| Clustering Method | K-means (155,823) | Louvain (736) |
|-------------------|------------------|--------------|
| prime minister    | 735              | 17,962       |
| press digest      | 3,089            | 4,066        |
| stories vouch     | 21,589           | 10,698       |
| verified stories  | 2,047            | 1,839        |
| reuters verified  |                  |              |

Table 2: Louvain detected 9 fine-grained clusters from the single cluster on government and social.

Figure 4: (top) The frequent terms using the default number of clusters $k = 250$. (middle) The frequent terms for true number of groups $k = 20$. (bottom) The results from Louvain. The number of detected groups $k = 14$.

The top panel of Figure 4 represents the scenario when an analyst has no prior knowledge about the data, as is the case with a new customer, so the number of intention groups is unknown. Using the default setting of k-means and $k = 250$ (meant for larger datasets), VIM displays the top frequent terms selected from the 33 largest clusters. Frequent terms appear in multiple groups without corresponding contextual terms, such as one, three, first time, new york, making it difficult to gain insight from such clusters.

The middle panel shows the second scenario when the analyst is familiar with the data and he or she estimates the number of groups to be the true number $k = 20$. The VIM pipeline completes the clustering task using k-means and displays the frequent terms from the top 5 largest clusters. Although there are common frequent terms in multiple clusters, other contextual terms and words that were important to each cluster surface so that analyst can at least identify three meaningful groups (Chinese government and the Olympic Games, American Idol show and Buddhist schools).

The bottom panel shows the results when VIM uses Louvain for community detection. The pipeline detects 14 groups from the 1,400 paragraphs. The detected groups are a good estimation of the true groups. Without prior knowledge of the dataset an analyst can now get the clustering result that is analogous to scenario two. In this example, we can still identify the Chinese government and Olympic Games, the American Idol and the Buddhist clusters.

5 CONCLUSION

In this paper, we have presented and open-sourced an important AI/ML component in a commercial data exploration and annotation tool used for labeling training data for building IVA intent classifier models. Our text clustering pipeline is flexible and provides optimal results with a minimum of user configuration (specify number of clusters or not, provided labeled sample for fine-tuning or not). The pipeline provides high quality clustering results and it is able to scale vertically on increasing text data volume and horizontally on increasing computation resources. We also showed the community detection techniques work well for text mining tasks. On our current configuration, analysts can get insightful clustering results in 15 minutes for the largest task evaluated. The clustering pipeline has greatly reduced the amount of data discovery effort required by the human analysts in our production application.
REFERENCES

[1] Ian Beaver and Abdullah Mueen. 2020. Automated Conversation Review to Surface Virtual Assistant Misunderstandings: Reducing Cost and Increasing Privacy. Proceedings of the AAAI Conference on Artificial Intelligence 34 (04 2020), 13140–13147. https://doi.org/10.1609/aaai.v34i08.7017

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008, 10 (2008), P10008.

[3] Xinyu Chen, Ian Beaver, and Cynthia Freeman. 2020. Fine-Tuning Language Models For Semi-Supervised Text Mining. In 2020 IEEE International Conference on Big Data (Big Data). 3608–3617. https://doi.org/10.1109/BigData50022.2020.9377810

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[5] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data (2019).

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019).

[7] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to information retrieval. Cambridge university press.

[8] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018).

[9] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for SQuAD. arXiv preprint arXiv:1806.03822 (2018).

[10] Ashwin Ram, Rohit Prasad, Chandra Khattry, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. arXiv preprint arXiv:1801.03604 (2018).

[11] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Semi-supervised Clustering for Short Text via Deep Representation Learning. CoNLL 2016 (2016), 31.