Intent Discovery With Or Without Labeled Data Using Dependency Parser

Anonymous Author(s)
Affiliation
Address
email

Abstract

In dialogue applications, machine learning classification models are often used to classify user utterances into different intents that help to understand the users. In real world scenarios, however, some utterances may not belong to any of the anticipated intent categories. Furthermore, supervised classification models are not a viable solution when data of a new domain is introduced without the corresponding labels. In this work, we present a clustering and evaluation approach that can be used in semi-supervised or unsupervised modes, depending on the (non-)availability of training data for new intent discovery. This method assigns meaningful intent-labels by determining the optimal number of clusters and evaluating the performance of the clustering results. In addition, it assigns a TF-IDF score to individual samples within a cluster.

1 Introduction

One of the goals in customer service dialogue applications is to automate the work of live agents while maintaining an exceptional customer experience. Therefore, accuracy in understanding the content of customers' utterances is crucial. The flexibility of natural language allows one meaning to be expressed in many different ways, each with different sets of words or phrases. Our task can thus be considered as finding the patterns that conform to this many-to-one relationship between textual form and meaning, thereby assisting the agent or Chatbot in taking the next action.

In what is the typical approach, we start by pre-defining a certain number of intents that cover the meaning of the most frequently occurring samples. We then label these samples with their correct intents and use them to train an intent classification model which can predict new unseen samples.

Given any new sample, then, one of the following scenarios applies: 1) it belongs to one of our pre-defined intents; 2) it belongs to one of our pre-defined intents, but it is not very similar to the training samples that belong to that intent; 3) it does not belong to any of our pre-defined intents, but it is very representative, and should therefore be considered as a new intent; 4) it does not belong to any of our pre-defined intents, but it does not appear very frequently in real world situations, and can therefore be simply labeled as OTHER.

While machine learning classification models have been shown to be effective in scenario (1), scenario (2) is more challenging than [1], and unsupervised clustering models can identify samples in scenario (3) to some extent (*e.g.*, samples belonging to large population clusters but classified as OTHER by a classification model might be considered as candidates for new intents). However, there is no guarantee that the samples in one cluster belong to one intent alone, and the algorithm can also not help with defining the name of the new intent for the cluster. Therefore, a comprehensive intent discovery method is needed to address the remaining three scenarios that are not handled by a traditional classification model.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Unsupervised intent labeling in dialogue environments has been studied in [2], showing how some context features, including POS tags and keywords, can achieve good clustering performance. However, in that approach, the features and intent-labels need to be manually selected, which may highly depend on the specific dataset. Multiple text clustering optimization methods have been explored as well, such as the Group-average Based Clustering method mentioned in [3], and the Maximum Entropy approach proposed by [4]. These approaches are effective when the amount of samples is static. However, in our application, new samples are added in frequently, which requires the implementation of methods that can assign an optimal label without frequently recomputing the number and composition of the clusters as the sample population changes.

A method for this task is ideal if it can answer three questions: 1) What is the optimal number of clusters?; 2) How to measure the performance of the clustering result?; and 3) How to label the clusters?

Deep learning algorithms typically do not rely on traditional feature-engineered NLP knowledge [5]. However, for applications where conventional classification models are ineffective (*e.g.*, the aforementioned problem which does not have a definite answer on how many categories we have overall), we reconsidered the contribution of traditional NLP models. These models, notably dependency parsing models, can provide syntactico-semantic information for each sentence, which can be used as an alternative labeling criteria to conventional manual labeling. This paper proposes a semi-supervised as well as an unsupervised clustering approach, based on dependency parsing model features, to discover the intents of new utterances. The proposed clustering method can answer the three aforementioned questions effectively.

2 Dataset

We conducted our experimental work on a dataset collected from a real world application currently in production, where standard redaction criteria replaced PII (Personal Identifiable Information) with tokens such as `<PHONE/>`and `<EMAIL/>`. These text dialogues were collected from both human-human and human-bot chats. All tokens were transformed to lowercase letters to reduce data sparsity further. Table 1 reports statistics of this dataset.

Table 1: Dataset description.

Name	Samples	Number of Intents
Training	8939	46
Test	2209	16 + OTHER

3 Models

We used three types of models: dependency parsing, Word2Vec and *k*-means clustering, all of them trained or obtained from publicly available Python packages. The Dependency parsing model was provided by Python package `spaCy`[6], where the model file used was `en_core_web_lg`. We trained two different Word2Vec models, using the method provided in `gensim`[7]. The first model was trained using single-word tokens generated by all the sentences in the training dataset, while the second model was trained using dependency parse triples of sentences as input tokens, where a triple consists of the elements of a dependency relation between word pairs in the sentence, namely the dependent word of the relation, the dependency relation label, and the head word of the relation.

We are particularly interested in studying the potential benefit of using triples as features because of the richness of information they provide, compared to single-word tokens. While they do not only double the amount of words, triples also unveil relationships between words, therefore we hypothesize that sentences sharing similar triples are more likely to share similar meaning than those simply sharing similar words. Consider the examples in Table 2: While Sentence 2 and Sentence 3 have different meanings, the single-word feature returns identical embeddings and the triple feature returns two different ones.

Table 2: Sentences and dependency triples.

<table>
<tr><th>Type</th><th>String</th></tr>
<tr><td>Sentence 1</td><td>‘unable to update software’</td></tr>
<tr><td>Triples</td><td>‘unable|ROOT|unable to|aux|update update|xcomp|unable software|dobj|update’</td></tr>
<tr><td>Sentence 2</td><td>‘need to study patients’</td></tr>
<tr><td>Triples</td><td>‘need|ROOT|need to|aux|study study|xcomp|need patients|dobj|study’</td></tr>
<tr><td>Sentence 3</td><td>‘patients need to study’</td></tr>
<tr><td>Triples</td><td>‘patients|nsubj|need need|ROOT|need to|aux|study study|xcomp|need’</td></tr>
</table>

The triples were formatted into a space-separated string, as shown in Table 2. K-means clustering models were trained using the cluster module in the `sklearn` [8] Python package with parameters set to: `random_state=23`, `n_init=10`, `max_iter=` 200.

4 Experiments

4.1 Semi-Supervised

Training Given that the number of intents in the training data is 46, we anticipate the optimal cluster number to be around this value. Therefore, we trained a series of clustering models with cluster number varying from 35 to 74. We are also interested in observing how different Word2Vec embeddings, whether trained with single-word tokens or triple tokens, contribute to the performance of clustering.

We used three different methods to convert the utterance samples into fixed-length vectors:

1) Apply the single-word token Word2Vec model to each word and average the resulting vectors into a single vector with 100 dimensions for each individual sample;
2) Apply the triples token Word2Vec model to each triple assembled from the dependency parser and average the resulting vectors into a single vector with 100 dimensions for each individual sample;
3) Average the vectors from 1) and 2) for each individual sample.

In total, the experiments produce 120 clustering results.

Evaluation Entropy measures the uncertainty of a random variable [9]. We adopted entropy as the metric to evaluate results, where the best performing clustering has the lowest entropy.

Let the random-variable ω_k represent the intents in the k-th cluster, then the entropy of a *cluster-group* is:

$$H(\omega_k) = -\sum_i p_{\omega_k}(i) \cdot \log_2 p_{\omega_k}(i), \tag{1}$$

where i is an intent-class.

Similarly, let the random-variable ϕ_i represent the clusters associated to the i-th intent, then entropy of an *intent-group* is:

$$H(\phi_i) = -\sum_k p_{\phi_i}(k) \cdot \log_2 p_{\phi_i}(k), \tag{2}$$

where k is a cluster.

The total entropy of a clustering was calculated by taking average values of all the clusters' individual entropy:

$$H_\omega = \frac{1}{|K|}\sum_{k\in K} H(\omega_k), \tag{3}$$

$$H_\phi = \frac{1}{|I|}\sum_{i\in I} H(\phi_i). \tag{4}$$

Table 3: Universal formula and examples.

Type	Value
Sentence 1	‘how do i change my email address on my account’
Sentence 2	‘i need to change the email address in my account’
Sentence 3	‘i want to change my email address on my account please’
Sentence 4	‘how can i change my email address for my account’
Key Tokens	‘nsubj:"i", dverb:"change", dobj:"address", pobj:"account"
Universal Formula	‘i-change-address-account’

4.2 Unsupervised

Universal Formula While in the absence of labeled data the training process is conducted in the same way, the evaluation process cannot calculate the entropy measurements without knowing the category of each sample.

In order to solve this problem, we have introduced the concept of *Universal Formula*, which uses patterns developed from key triples to extract the main syntax and semantics structure of the sentences, thus allowing us to bring certain samples that share the same pattern into the same category. Generally speaking, direct verb[1], main subject[2], main direct object[3] and main indirect object[4] are considered to convey the main meaning of a sentence, as they are taken from dependency relations in the main clause of the sentence; and if triples related to these tokens from a sample are provided, we could briefly infer the intent of the sample.

Here, we define triples with dependency type of "ROOT"(root word), "nsubj"(subject), "dobj"(direct object) and "pobj"(indirect object) as key triples. We take all the key triples from a sample, extract the key tokens and reconstruct them into string format, which is then considered as the substitute of the intent label. Table 3 shows examples of sentences that share the same universal formula value, *i-change-address-account*.

Evaluation Entropy was calculated in the same way as in Section 4.1, except that the universal formula of sentences provides the alternative intent labels.

5 Discussion

5.1 Semi-Supervised

Figure 1 shows the entropy value variation of clustering model run with different cluster numbers. Cluster-group entropy H_ω tends to decrease with the increased number of cluster, while intent-group entropy H_ϕ behaves the opposite way. This is consistent with the observation that the intent-group entropy approaches zero (deterministic) as the cluster number decreases to one, and that the cluster-group entropy approaches zero (deterministic) as the cluster number increases to match the number of samples.

Considering that H_ω and H_ϕ compete against the cluster number, we first normalized the two entropy values into $[0, 1]$ based on the minimum and maximum values in each feature group, then took the average value of the two to obtain a combined measure of the performance of the clustering as shown in Figure 1.c. The better performing clustering, *i.e.*, the ones with the lowest entropy, occur at cluster number 35, 43 and 53. We observed that the triple feature overall performs better among the three methods explored. We selected the K=53 as our best option because it provisions some cluster space for potential new intents in the future.

[1]Verb directly connected to main subject or object or root verb, or sometimes the root verb itself is the direct verb.

[2]Subject that is closest to the root verb.

[3]Object that is closest to the root verb and directly connected to a verb.

[4]Object that is closest to the root verb and indirectly connected to a verb, *e.g.*, via "in" or "at".

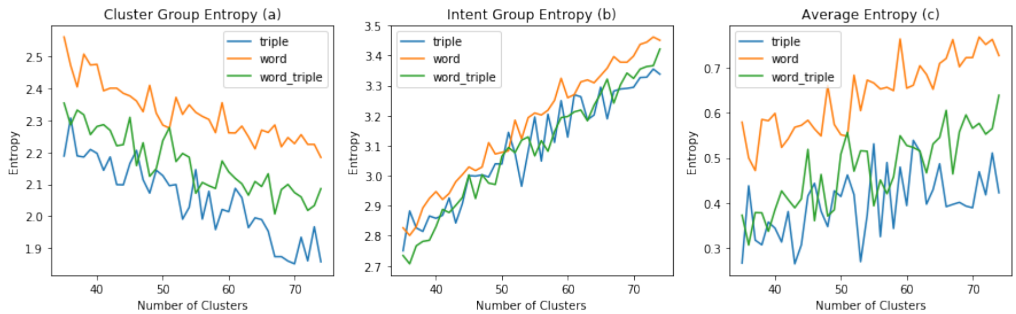

Figure 1: Entropy of clustering calculated using intent labels.

5.2 Unsupervised

In the absence of labeled data, we need an alternative way to measure the entropy. The universal formula of sentences presented in Section 4.2 can generate a certain amount of sentence patterns that allow unlabeled samples to be grouped. The intuition is that samples belonging to the same pattern group should be assigned together in the clustering result, therefore these pattern labels can be effective for entropy calculation in place of actual intent labels. As expected, we observed how the average entropy of the pattern groups increases with the number of clusters increasing, as Figure 2.b shows. We repeated the same step as in Section 5.1 to get the average value of the entropy and the results of overall entropy are shown in Figure 2.c. Notice how the triple feature performs slightly better than the other two sets of features, and how the lowest entropy value occurred on cluster number 56, which is in the vicinity of the previous result obtained with intent labels, namely 53.

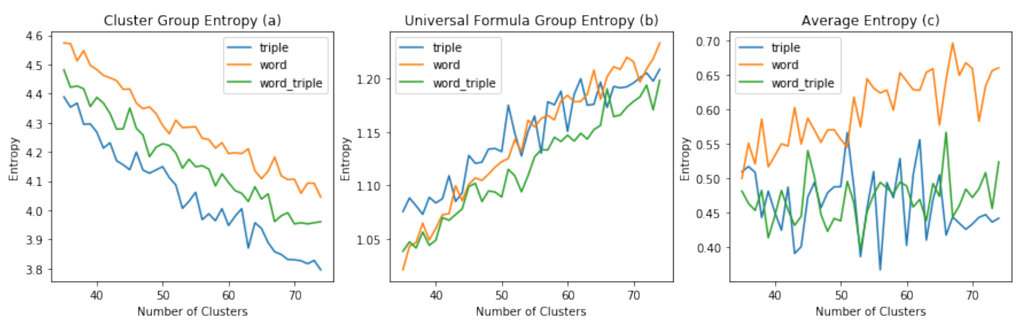

Figure 2: Entropy of clustering calculated using universal formulas.

To further compare the two results, Figure 3 plots the two lines of triple-feature clustering results from Figure 1c and Figure 2c. The strong correlation suggests that the universal formula can be effectively used as an alternative way to evaluate the clustering performance as well as to obtain the best cluster number, in the absence of labels.

5.3 Labeling Clusters and Ranking Samples

The proposed method relies on the observation that samples belonging to the same cluster tend to share similar triples. Generally speaking, triples with dependency type as "ROOT"(root word), "nsubj"(subject), "dobj"(direct object) and "pobj"(indirect object) convey the main meaning of a sentence and if triples of these types are provided, we could briefly know the intent of the sample.

We calculate the document frequency [10] of the four dependency types of triples in each cluster and pick the top candidates as the cluster label components. Although samples in the same cluster are usually similar to each other, not all the samples are equally relevant to the core meaning of the cluster they belong to. The task of intent discovery, however, is not concerned with the meaning of all the utterances, it is instead concerned with the meanings of most frequent utterances. To retrieve the most relevant samples that have the core meaning of a cluster, a TF-IDF [10] method was used to

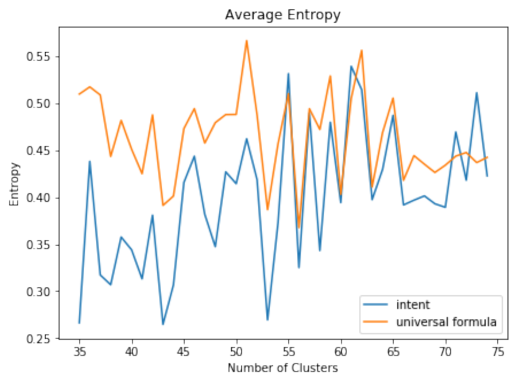

Figure 3: Entropy measured using intent labels and universal formulas.

calculate a relevance score for each sample as the summed value of all the TF-IDF scores for all the triples in the sample that belong to the four categories as mentioned above.

We reserved some intent categories in our test samples and did not include those ones in our training data. To better understand the clustering performance in terms of grouping and finding new intents, we predicted test samples using clustering model trained with cluster number 56, as we obtained in the unsupervised process. The results are shown in Table 4.

Table 4: Partial results from test dataset.

Intent Name	Recall	Cluster Label
ISSUE_WITH_PRODUCT	0.33	[‘we-need’, ‘download-is’, ‘i-pressing’, ‘it-remove’, ‘having-problem’, ‘got-virus’, ‘install-deluxe’, ‘with-download’, ‘at-risk’]
CHANGE_ADDRESS	0.75	[‘i-do’, ‘address-is’, ‘i-changed’ ‘have-access’, ‘changed-address’, ‘change-key’, ‘on-account’, ‘from-email’, ‘to-info’]
ISSUE_FIXED	0.50	[‘have-computer’,‘need-assistance,‘have-problems’, ‘that-seems’,‘i-think’,‘that-worked’, ‘i-let’,‘you-know’,‘restart-computer’]

5.4 Agent in the Loop

The intent discovery process relies on human input (agent) in a few steps. First, the agent helps with completing the intent labeling of the cluster. That is, agents read through the list of phrases provided by the automatically generated labels from each cluster and come up with an intent name that conforms with the format convention prescribed by existing intents. Second, the agent verifies if the highly ranked samples in each cluster indeed have the meaning as the label of the cluster by answering "yes" or "no" on those samples. Third, when an intent classification model is available, the agent runs clustering on samples that are classified as OTHER by the model.

As mentioned in the second scenario of Section 1, we are interested in finding any samples missed by the model as this is an indication that they actually belong to one of the known intents. To accomplish this, the agent verifies if any of the cluster labels share similar meaning with the known intents, deciding when clusters should be merged into the training data for intent classifier to improve future models.

In the absence of labeled data, on the third step the agent reviews all the cluster labels and verifies if any of them can be merged due to semantic proximity.

6 Conclusion and Future Work

This paper proposed a method to discover unseen intent categories along with intent labels that are meaningful. The method addresses the case where samples consist of utterances classified as OTHER by an intent classification model and the case where utterance samples do not have labels assigned at all. It helps with discovering samples that were mis-classified as OTHER, samples that fall into a large population cluster that should be assigned with a new intent label, and samples that fall into a smaller population group that can be simply labeled as OTHER based on interest of the application.

This approach processes utterance samples with a dependency parser to create triples that are used as tokens in a Word2Vec embedding. With the performance criterion set to the average entropy of cluster groups and label groups, the method is able to find the best cluster number. We show experimentally how the Word2Vec embedding from dependency triples outperforms the standard Word2Vec embedding from single word tokens. The most common triples from each cluster provide an effective intent label to the clusters, while TF-IDF ranks each cluster sample based on the frequency of the triples. Finally, the method explains the role of an agent in the loop when available.

Broader Impact

Intent discovery methods with a minimum of required manual work required are extremely helpful for researchers to explore classification of new utterances or any other intent analysis related activities. As long as data does not have any confidential or personal content, this application should not lead to any harm. The results could be not as satisfactory, if the data baseline keeps changing, *e.g.*, utterances with completely new intents are added very frequently, however, if the system is deployed in a stable platform for a specific client, this situation should rarely happen; bias could be leveraged in identified samples, which is simpler and grammarly regular samples would be captured more easily, however, with larger collection of training samples added and more complex Universal Formula for sentences applied, the bias would be reduced accordingly.

References

[1] Bahman Zohuri and Masoud Moghaddam. Deep learning limitations and flaws. 01 2020.

[2] Chen Shi, Qi Chen, Lei Sha, Sujian Li, Xu Sun, Houfeng Wang, and Lintao Zhang. 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, October-November 2018. Association for Computational Linguistics.

[3] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, Lansdowne, VA, USA, February 1998. 007.

[4] Cheng Niu, Wei Li, Rohini K. Srihari, Huifeng Li, and Laurie Crist. Context clustering for word sense disambiguation based on modeling pairwise context similarities. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 187–190, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[5] Yang Jiang, Nigel Bosch, Ryan Baker, Luc Paquette, Jaclyn Ocumpaugh, Alexandra Andres, Allison Moore, and Gautam Biswas. *Expert Feature-Engineering vs. Deep Neural Networks: Which Is Better for Sensor-Free Affect Detection?*, pages 198–211. 06 2018.

[6] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[7] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher,

M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[10] Christopher Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.