Intent Recognition

Anonymous EMNLP submission

Abstract

Intent Recognition is the form of Natural Language Processing concerned with AI. Intent Recognition is a sequence classification task where the model classifies what the users want to achieve. In this paper we present our findings of comparing classical machine learning algorithm, Support Vector Machine, and BERT: Bidirectional Encoder Representations from Transformers.

1 Introduction

A Dialog System mainly consists of six components; Automatic Speech Recognition (input from the user), Speech/Natural Language Understanding (SLU/NLU), Dialog State Tracker, Dialog Policy, Spoken/Natural Language Generation (SLG, NLG), and Text to Speech (Output to the user). In this article, we will focus on the NLU component.

Natural Language Understanding component, as the names suggests, is the unit that understands what the user wants. It converts the user input into semantic representation that is used by the dialog state tracker. NLU component has many functions and one of the most important functions is to recognize intent of the user. We will call this task intent recognition or intent classification.

2 Related Work

Work on classification of short text has been done for years. Classifying text based on sentiment, intent and entity can be done using classic machine learning algorithms like Naive-Bayes and Bag-of-Words(BoW) technique. But these algorithms are highly dependent on the type of features used and the vocabulary of the training set (Wang and Manning, 2012). To circumvent this we can use Support Vector Machines (Cortes and Vapnik, 1995).

When it comes to modern deep learning algorithms, Convolutional Neural Networks (Hashemi et al., 2016) and Long-Short Term Memory Unites (Meng and Huang, 2017) have been applied directly on intent classification datasets using pre-trained feature vectors. These have worked better than the classical machine learning algorithms because they are able to catch long-term dependencies. Nowadays, transformers such as BERT (Devlin et al., 2019) are used for various NLU and NLI tasks including intent classification (Chen et al., 2019). The work done for this paper is similar to (Chen et al., 2019). The only difference is that we don't experiment on slot filling data.

3 Model Architectures

Below are the two model architectures we experimented on:

3.1 Support Vector Machine

Support Vector Machine is a learning algorithm that determines the best decision/classification boundaries between vectors that belong to a given category. This algorithm can be applied to any kind of data if that data is first converted to a vector representation. These vector representations are used as features.

3.2 BERT: Bi-directional Encoder Representation Transformer

BERT is a a large pre-trained multi-layered transformer based (Vaswani et al., 2017) encoder developed by Google. It is pre-trained using a plain-text corpus and then it is fine-tuned on downstream tasks like sentiment analysis, entity and intent recognition, etc. The BERT model architecture used in this experiment looks as follows:

- 12 Layers.
- 768 Hidden Units.
- 12 Attention Heads.

Model	Accuracy	F1-Score
SVM	98.25	90.78
BERT	99.375	95

Table 1: Evaluation on Test Set

Intents	SVM	BERT
atis-flight	8	4
atis-flight-time	All Correct	All Correct
atis-airfare	2	All Correct
atis-aircraft	1	All Correct
atis-ground-service	All Correct	All Correct
atis-airline	3	1
atis-abbreviation	All Correct	All Correct
atis-quantity	All Correct	All Correct

Table 2: Shows which intents were predicted incorrectly

4 Dataset

The dataset we used in this experiment is the famous ATIS (Guo et al., 2014) dataset. It is used widely in NLU research. The training set contains 4351, validation set contains 483, and test set contains 800 examples. There are a total of 7 intents; flight, fare, airline, ground service, abbreviation, aircraft, quantity, flight time.

5 Experiment

SVM was run on CPU and BERT was run on Google Colab's Nvidia Tesla T4 GPU. Features used for Support Vector Machines are tf-idf vectors. TF-IDF is a numerical statistic that tells the importance of a word in a given sentence. TF-IDF scores are converted to vector representations in a matrix and then fed into SVM. Loss used for SVM is squared hinge loss. The model is trained for 1000 iterations.

The input representation for BERT are concatenation of wordpiece embeddings and positional embeddings. Maximum length of sequence for BERT is 128. Fine-tuning is done on batch size of 16, learning rate is 3e-5 with cross entropy loss for 3 epochs. For automatic evaluation, we use Accuracy and F1 for both the models.

Manual error analysis is done on the test set to see which intents were incorrectly predicted and why.

Intents	Number of Examples
atis-flight	632
atis-flight-time	1
atis-airfare	48
atis-aircraft	9
atis-ground-service	36
atis-airline	38
atis-abbreviation	33
atis-quantity	3

Table 3: Count of each intent in Test set

6 Result

Table 1 shows that BERT outperformed SVM by few decimals in the accuracy metric. And for F1 score, BERT outperformed SVM by almost 5 percent. This was expected as BERT uses Masked Language Modeling which is deeply bi-directional.

7 Error Analysis Discussion

Table 2 tells us that which of the eight intents were classified incorrectly and how many times in both the models. SVM in total made 14 errors and BERT made 5 errors. We can see that BERT is correct 100 percent in 6/8 intent categories and SVM is correct in 4/8 categories. Division of intents in the test set are shown in table 3. By looking at both these tables, we can tell that the reason *atis-flight* has been incorrect most of the times in BERT and SVM is because *atis-flight* has the most number of examples.

Both SVM and BERT, misclassified *atis-flight* text to *atis-aircraft*, *atis-quantity* or *atis-airfare*. The reason could be that the utterances, which were generated by the user, are questions related to flight but have the words *aircraft* in the later part of sentence, *how many* in the start of the sentence, or *fares* in the later part of sentence. But the confusing thing is that some of these utterances have the word *flight* in them as well and our model still misclassified them. *atis-airline* was misclassified as *atis-flight* by both the models in the sentences which had the word *flight* or *flies* in them. SVM misclassified *atis-airfare* to *atis-flight*, and *atis-aircraft* to *atis-quantity*.

The tables above do not tell us about the misclassification overlap of BERT and SVM. 4/8 sentences misclassified by SVM were misclassified by BERT in the *atis-flight* intent. BERT and SVM have one misclassification in common of *atis-airline* intent.

We will look at one interesting example of each of these.

First, let's consider a sentence from *atis-flight* category. The sentence is **List delta flights from seattle to salt lake city with aircraft type.** This was misclassified by both to *atis-aircraft*. Now this query can be classified as a question related to aircraft or a question related to flight. The reason it belongs to *atis-flight* is because usually, when asked about the details of a flight, type of aircraft is mentioned in the flight details. So we cannot entirely blame our language models if the question is ambiguous. Our data is not hierarchical. If the data was hierarchical then we can blame our model for giving us a misclassification. This also shows that SVM and BERT have difficulties in finding hierarchical patterns in the text.

Now, the second example is from the *atis-airline* category: **What airlines off from love field between 6 and 10 am on june sixth**. BERT misclassified this to *atis-flight-time* and SVM misclassified this to *atis-flight*. BERT misclassified this sentence because there is a time and date mentioned in the end of the sentence. This is interesting because it shows that BERT somehow understood the concept of date and time. But it is incorrect nonetheless. What we do not understand is why SVM misclassified this sentence to *atis-flight*. Our hypothesis is that SVM got confused between *field* and *flight*.

8 Conclusion

In this paper, we built and tested two machine learning models and ran our own error analysis. We found out that both the models give very good test accuracy. Our error analysis shows us that both SVM and BERT have difficulties understanding order and hierarchy in the text. What we can do to nullify this problem is to either build multi-label multi-class datasets which have labels like *flight-fare* or we build hierarchical datasets.

References

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. In *Machine Learning*, pages 273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Daniel Guo, Gokhan Tur, Wen-tau Yih, and Geoffrey Zweig. 2014. Joint semantic utterance classification and slot filling with recursive neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 554–559. IEEE.

Homa B Hashemi, Amir Asiaee, and Reiner Kraft. 2016. Query intent detection using convolutional neural networks. In *International Conference on Web Search and Data Mining, Workshop on Query Understanding*.

Lian Meng and Minlie Huang. 2017. Dialogue intent classification with long short-term memory networks. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 42–50. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sida I Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94.