Sequential Labeling for Building Sensors

Zheng Luo
Department of Computer Science
University of Virginia
zl5sv@virginia.edu

Introduction

Cyber physical systems, which will potentially be rapidly developed and deployed in near future years, have a large demand on sensors. One application of cyber physical systems is the smart building which monitors every detailed state of itself based on a large set of sensors. Since each sensor can have different functionality and manufacture, properly recognizing and managing those sensors becomes an issue that must be solved. In this project, we will illustrate a solution using conditional random filed (CRF) [1] to perform sequential labeling towards texts that come with sensor point names to identify different sensors types, locations and other attributes. We will develop a query strategy for active learning in order to save as large human efforts as possible.

Related Works

A recent research [2] has conducted a clustering-based active learning method to classify different sensor types from point name texts/features. The approach maximizes the accuracy of classification while minimizing the required size of training set at the same time. The purpose is to decrease the human efforts as much as possible for labeling and recognizing the large amount of sensors. This approach only considers the sensor types while point names contain other important information including building names, room numbers, and other attributes. In order to make use of the full information of sensor point names, we further conduct sequential labeling for those sensors to better understand smart buildings.

Another work [4] uses transfer learning to classify buildings which are unseen previously. The basic idea is to use a set of different classifiers trained based on several buildings as a set of different perspectives. When applying those pre-trained classifiers to new buildings, the classification results are weighted combinations of them. Therefore, weights are learned during the transferring process. This work builds on the research [2], so only sensor types are considered. As full information of point names is labeled, this transfer learning approach is supposed to be updated to take full sensor information into account.

Research Plan

The first step of this project is to process the dataset of sensor point names. We use beginning-inside-outside (BIO) format to tag the sensor point names where each point name is a text string containing sensor type, location, and other attributes such as identification number. By using the BIO tagger, we will be able to convert those text strings into a full labeled dataset. This task is a good start to get familiar with the input/output of this ongoing project.

The second step is to take part of the dataset as a training set to train a conditional random filed. The sequential labeling will be performed on the rest of the dataset to access the performance. Furthermore, we would like to perform active learning in a structured learning problem. The model for parsing the input strings will still be a CRF but the query strategies can be different. The purpose of active learning is that we want to achieve a good recognition performance using as few human efforts as possible. After this task, we will combine active learning and transfer learning to annotate new buildings where we assume that the new buildings are unknown but have been embedded with similar sensors that have comparable point name structure.

Baseline Implementation

Since there are a series of studies using CRF for shallow parsing, we want to use current existing implementations to be our baselines. Specifically, for active learning, we will refer to query strategies mentioned in [5]. Our current dataset is a set of strings with a fixed length 14. Each character in each string

is manually labeled. For example, the point name string "SODA4337A_RVAV" is labeled as "B-site, I-site, I-site, B-ahu, I-ahu, B-room, I-room, I-room, I-room, O, B-vav reheat discharge air pressure sensor (i.e. RVAV in short), I-RVAV, I-RVAV, I-RVAV". A CRF model can be easily trained by a set of such labeled instances. The model will be able to correctly label similarly strings/patterns in a test set. Our metric for performance is "phrase-level" accuracy, that is, it is considered as a correct label only when a whole pattern is correctly labeled. For example, SOD is considered correctly labeled if it is labeled to be a whole pattern as "B-site, I-site, I-site", and the labels for this pattern is cut off at the location of this pattern; that is, as an illustration, it is incorrect if "SODA" is labeled as "B-site, I-site, I-site, I-site".

For active learning, we need to "ask" the model when training it, such that the most information-rich samples can be picked out as early as possible. In other words, we can achieve a high accuracy with as few training samples as possible by proper query strategies. For the baseline implementation, we use only a few samples as an initial training set to train a CRF. This CRF will be applied to the test set without knowing the ground-truth labels, and output a sequential labeling confidence (i.e. the probability of current labeling output). Then we look at the "unlabeled" (in order to perform active learning, we assume the training samples except those in the initial set are unlabeled) samples, find out the samples which have the same or similar features compared to the samples in the test set with the lowest labeling confidence. These picked out samples, together with there manual labels will be further used to train the CRF again. This algorithm is shown as the following Algorithm 1.

Initialize a **training set** with only few samples.
Train a CRF model **crf**.
While size(**training set**) < maximum number allowed:
    Obtain **test_confidence** by applying **crf** on the **test set**.
    Find a sample **sample1** with low confidence and is representative in the **test set**.
    Find few samples **set1** which have similar features (by edit distance) as **sample1** in the **unlabeled set**.
    Label and add **set1** to **training set**, and remove **set1** from **unlabeled set**.
    Retrain **crf** using the current **training set**.

Algorithm 1: Our Baseline for Active Learning

The size of **set1** is called batch size and can be user defined. In our experiments, we keep the batch size being 1. When finding a sample in the test set, we want balance between confidence and representative. For example, if a sample results in the lowest confidence but its features are very rare in the test set, we need to ignore this sample and choose the one that results in a relatively low confidence but whose features/patterns are very common in the test set. The can be done by clustering the test set based on edit distance, and weighting the confidence by its cluster size and distance to the center of its cluster. It is also possible to make use of the confidence on the **unlabeled set** to find **set1** (this could be updated later). We use 10-fold cross validation to access the baseline active learning results. The following Figure 1 shows the results compared to the query strategy that uniformly taking samples from **unlabeled set**. For comparison purpose, even though it is not needed, the uniform case also has the same initial training set of size 2 as the query by confidence. In the figure, the two plots both start at number 3 (i.e. when the first new sample added to the training set).

Active Learning with Partial Human Labels

It is easy to notice that the phrase-level errors are made due to some unseen patterns of a CRF. Even though we have only few training samples, it is unlikely that the labels for a whole string are all incorrect because some patterns like the building name are extremely common to appear. On the other hand, the previous algorithm, which uses the confidence on the whole string, may keep the CRF model away from some important features because the confidence is averaged. Therefore, when performing active learning, we want to obtain the "confidence" for each phrase instead of the whole string. This can be done by outputting the marginal distributions of each label node in the CRF graphical model. For this purpose, we will be using the toolbox provided by [6].

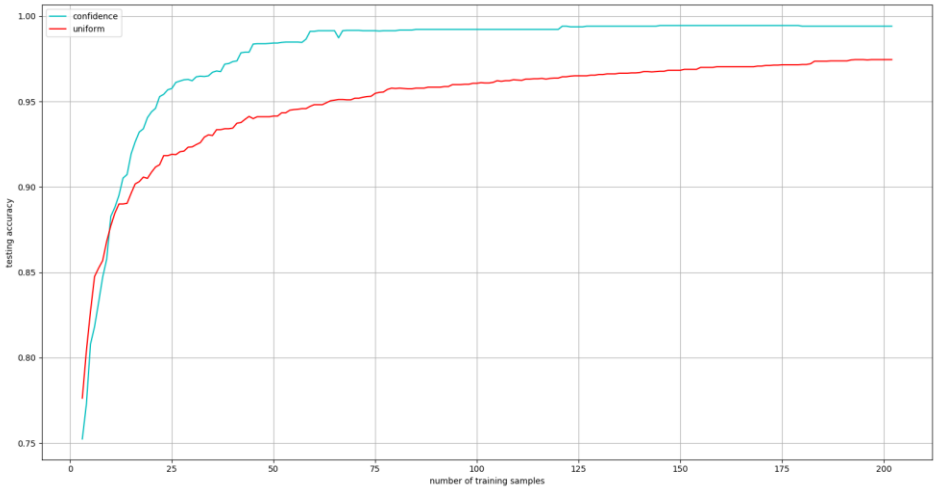

Figure 1: Performance of Baseline Active Learning

Given the pattern confidence of samples in the test set, we can look for patterns with low confidence, and find such patterns in the "unlabeled" set, and then pick out strings having these patterns and only label the patterns. The unlabeled part will be pseudo-labeled by the current trained CRF model in order to use as less human efforts as possible on labeling. In fact, this strategy will only focus on the patterns existing in the test set and will not care about the accuracy of labeling patterns that doesn't exist in the test set. Therefore, we need to carefully handle the pseudo-labeled part which may lower down the performance on the test set, and make a balance between pseudo labels and human labels. The idea of this strategy has been written as the following Algorithm 2. It is also possible to update the query strategy such that pseudo-labeled patterns in the training set can be revisited and manually handled if necessary.

Initialize a **training set** with only few samples.
Train a CRF model **crf**.
While size(**training set**) < maximum number allowed:
    Obtain **test_patten_confidence** by applying **crf** on the **test set**.
    Find a pattern **pattern1** with low pattern confidence and is representative in the **test set**.
    Find few samples **set1** which contain **pattern1** in the **unlabeled set**.
    Manual-and-pseudo label and add **set1** to **training set**, and remove **set1** from **unlabeled set**.
    Retrain **crf** using the current **training set**.

Algorithm 2: Active Learning with Partial Human Labels

There is also an unknown behavior that needs us to test. In general, labeling sequence by sequence means a whole sequence is "decoded" together at a time. Therefore, the assumption behind the CRF model is that the grammar (how people construct the point name strings) has impact on each transition in the strings. We assume that point name strings are constructed in a certain manner, that is, patterns in strings has some transition relationships. This may be true for the strings made by a certain manufacture; but it is still helpful to figure out how much impact this assumption can have because we will generate pseudo labels for some patterns to train a CRF.

Timeline

This rotation project is considered as a part of data mining project for smart buildings. The dataset preprocessing, baseline implementation, and other experiments related to CRF modeling and query strategies

were finished during several weeks of the second rotation. The unfinished part, which is the active learning with partial human labels, will be continued in the third rotation.

Future Research

Conditional random filed is a structured prediction method suitable for sequential labeling task. It has outstanding performance on many problems of text mining and natural language processing. In recent years, it has still been gradually improved by modifying its probabilistic structure and combining with other language models such as recurrent neural networks to improve representative power. For example, some new innovations, such as embedded latent CRF [3], use long short-term memory (LSTM) to enrich embedded representations. This type of techniques could be used or referred in future researches associated with large document or text datasets.

Reference

[1] Lafferty, J., McCallum, A. and Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[2] Hong, D., Wang, H. and Whitehouse, K., 2015, October. Clustering-based active learning on sensor type classification in buildings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 363-372). ACM.

[3] Thai, D., Murty, S., Bansal, T., Vilnis, L., Belanger, D. and McCallum, A., 2017. Low-Rank Hidden State Embeddings for Viterbi Sequence Labeling. arXiv preprint arXiv:1708.00553.

[4] Hong, D., Wang, H., Ortiz, J. and Whitehouse, K., 2015, November. The building adapter: Towards quickly applying building analytics at scale. In Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (pp. 123-132). ACM.

[5] Settles, Burr, and Mark Craven. "An analysis of active learning strategies for sequence labeling tasks." Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 2008.

[6] Sutton, Charles. "GRMM: GRaphical Models in Mallet." http://mallet.cs.umass.edu/grmm/. 2006.