Weak Supervision for Sequence Modeling

Shiying Luo

Advisor: Stephen Bach

May 15, 2019

Abstract

Recent methods for aggregating weak supervision sources have succeeded in modeling the accuracies and correlations among all sources using generative models, but most of them are applied to classification problems such as relation extraction. In this project, we applied the weak supervision modeling scheme to sequence modeling, more specifically, the Name Entity Recognition problem.

1 Introduction

Complex supervised models such as deep neural networks have achieved remarkable results in recent years. However, such models require massive amount of labeled data in training, and obtaining such magnitude of training data is expensive and time-consuming. To overcome the bottleneck of obtaining training data, there is growing interest of weak supervision, supervision that is noisier but cheaper to obtain. Examples of weak supervision include heuristic labeling rules, knowledge bases, and crowdsourcing.

In this project, we follow the scheme of the Snorkel[1] system by writing labeling functions to generate labels and use a generative model to aggregate votes by all labeling functions. Because these supervisions often conflict with each other, a generative model is usually trained in order to aggregate the labels from different sources to induce a single set of "reference standard" consensus labels. The labeling functions also has the option of abstention if it does not have any information regarding the underlying label. The Snorkel scheme and other similar work assume a constant precision for each labeling function across the entire corpus. In our project, all labeling functions are written to label one specific class. As a result, instead of assuming consistent precision, we assume a different precision for different underlying class for different labeling function.

We focus specifically on the named entity recognition problem in this project. The same way of modeling can be easily extended to other sequence modeling problems and other weak supervision sources such as crowdsourcing.

2 Related Work

Name-entity recognition (NER) is an important sequence labeling task in natural language processing. Current state-of-the-art methods for NER [3] [4] focus on complex neural architectures such as LSTM but deep learning methods usually only perform well when there are large amount of training data. There is another stream of work that focuses on discriminative models such as Maximum Entropy Models [5] and Conditional Random Fields(CRFs) [6]. These models can also achieve strong predictive performance but they are not well suited for label aggregation. In addition to the supervised setting, previous work[2] has studied unsupervised Hidden Markov Models(HMMs) in the crowdsourcing setting. Their model is similar to our work in trying to aggregate noisy labels, but our work also focuses on how to write effective labeling functions and use class-dependent precision in the generative model.

There is also work that modifies the source consistency assumption used in generative models to aggregate noisy votes. For example, the Socratic learning paradigm[7] augments the structure of the generative model by using feedback from a corresponding discriminative model to automatically identify the subsets in the training data in which the supervision sources perform differently than on average. Our work is different in leveraging the structure of generative model and not using any information from downstream discriminative model. The REHESSION framework [8] identifies each labeling function's proficient subsets based on context information and only trusts labeling functions on these subsets. However, the contexts of named entity tags are not well defined as in relation extraction problems and we need a different way of modeling precision.

3 Methods

3.1 Naive Bayes Model with Consistent Precision

The first model we build is a Naive Bayes model that assumes each labeling function has consistent precision over the corpus. Instead of hand labeling data, we write labeling functions to express various weak supervision sources such as patterns, heuristics, external knowledge bases etc. A table of all labeling functions used can be found in Table 1. The precision and recall are calculated using a small development set to provide a general idea of the performances of the labeling functions and are not used in the generative model.

Table 2 gives an example of how labeling functions work. The sentence "U.N. official Ekeus Smith heads for Baghdad" is first tokenized. In this example, "U.N" is an organization(ORG), "Ekeus Smith" is a person(PER), and "Baghdad" is a location(LOC). Labeling function No.12(See Table 1) first finds consecutive capitalized letters in the sentence and then tries to find key words such as "official", "launch" etc in the context of them. In this sentence, both "U.N." and "Ekeus Smith" are matches for this labeling function and they are labeled as "ORG". For the other tokens, this labeling function does not have any information so it abstains(ABS).

We assume that tokens are independent of each other. Let $\{1, 2, \ldots, m\}$ denote the set of underlying classes. Let n be the total number of labelling functions and the outcome from the ith labelling function is denoted by L_i, $i = 1, 2, \ldots, d$. Each L_i takes value in $\{1, 2, \ldots, m\} \cup \{ABS\}$, where ABS stands for abstention. We impose the standard assumption that given the label of a token, the outcomes from

ID	Target	Labeling Rule	Precision	Recall
1	PER	DBpedia exact matches	96.35%	40.06%
2	PER	Consecutive capitalized words + Context words(Mr, talk, who...)	47.68%	20.14%
3	PER	Consecutive NNP/NNPS + Context words(Mr, talk, who...)	31.29%	84.96%
4	LOC	DBpedia exact matches	54.05%	89.29%
5	LOC	Consecutive capitalized words + Key Last word(Airport, City)	60.25%	2.63%
6	LOC	Consecutive NNP/NNPS + Key Last word(Airport, City)	63.15%	2.68%
7	LOC	Consecutive capitalized words + Context words(eastern, arrive...)	55.41%	17.47%
8	LOC	Consecutive NNP/NNPS + Context words(eastern, arrive...)	57.46%	17.25%
9	ORG	DBpedia exact matches	62.06%	6.89%
10	ORG	Consecutive capitalized words + Key word(University, Ltd..)	51.62%	8.49%
11	ORG	Consecutive NNP/NNPS + Key word(University, Ltd...)	53.15%	7.73%
12	ORG	Consecutive capitalized words +Context words(official, launch...)	15.47%	8.42%
13	ORG	Consecutive NNP/NNPS + Context words(official, launch...)	16.85%	7.81%
14	MISC	Country adjectives matches	65.20%	38.61%
15	MISC	Consecutive capitalized words + Key Last word(Cup, Open...)	82.35%	7.92%
16	MISC	Consecutive NNP/NNPS + Key Last word(Cup, Open...)	81.48%	7.47%
17	O	Not capitalized words	96.37%	83.37%
18	O	Not NNP/NNPS	97.82%	81.86%

Table 1: All Labeling Functions Used

labelling functions are conditionally independent. Each labelling function can reject to label a token. For the i-th labelling function, we denote the abstention probability by β_i. We also assume that the i-th labelling function has precision α_i if it does not abstain from labeling the token. There are three possibilities of the i-th labeling function:

1. If the labeling function makes the right label

$$P(L_i = y|Y = y) = (1 - \beta_i)\alpha_i$$

2. If the labeling function makes the wrong label

$$P(L_i \neq y, L_i \neq ABS|Y = y) = (1 - \beta_i)(1 - \alpha_i)$$

3. If the labeling function abstains

$$P(L_i = ABS|Y = y) = \beta_i$$

For any given token, we can write the likelihood of observing all labels given by the labeling functions as

$$P(L_1, L_2 \dots L_n) = \sum_{y=1}^{m} P(Y = y) \prod_{i=1}^{n} P(L_i|Y = y).$$

Sentence	U.N.	official	Ekeus	Smith	heads	for	Baghdad
True Tags	ORG	O	PER	PER	O	O	O
LF12 Labels	ORG	ABS	ORG	ORG	ABS	ABS	ABS
LF17 Labels	ABS	O	ABS	ABS	O	O	ABS

Table 2: Example of Labeling Function Outputs

Since all tokens are assumed to be independent, the total likelihood is the product of such probabilities from each token in the text.

We train a Naive Bayes classifier with this likelihood and use gradient descent to get estimates of all parameters.

3.2 Hidden Markov Model with Consistent Precision

The first method assumes that tokens are independent of each other. However, in sequence modeling, this is hardly true. For example, if one token is labeled as organization, the token that follows it will very likely be organization.

Therefore, instead of a Naive Bayes Model, we experiment with a Hidden Markov Model. Essentially, we introduce a transition matrix into the model and incorporate the transition probability into the likelihood calculation.

3.3 Naive Bayes Model with Class-Dependent Precision

Since each of the labeling function is designed to label only one class, we modify the consistent precision assumption and have a different precision parameter for each underlying class for each labeling function.

4 Experiments

4.1 Datasets

For testing and evaluations, we use the development set of the CoNLL-2003 dataset(English). The dataset has 3,466 sentences and 51,362 tokens.

The data contains entities of four types: persons (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC). In total, there are 1842 persons, 1837 locations, 1341 organization, and 922 miscellaneous names. The IO tagging scheme is used, so we have 5 classes: O, PER, ORG, LOC, and MISC.

4.2 Results

We evaluate our model precision, recall and f1 score, and all f1 scores can be found in Table 3.

	Majority Vote	NB Single Precision	HMM Single Precision	NB Class Precision
Person F1	0.61	0.42	0.22	0.62
Location F1	0.84	0.74	0.32	0.82
Organization F1	0.21	0.16	0.06	0.22
Miscellaneous F1	0.65	0.54	0.08	0.64
Overall F1	0.64	0.51	0.19	0.63

Table 3: F1 Score for All Models

4.3 Discussion

The Naive Bayes(Single Precision) Model does not do very well because the consistent precision assumption is particularly not right in our case. The labeling function is defined to vote only one class, so it should have very different precision for that class and all other classes. Observe that when we switch to the Naive Bayes(Class Precision) Model, the scores all improve.

The HMM(Single Precision) Model performs very poorly because by modeling the transition matrix, the probability of transitioning from "Other" to "Other" is so large that a lot of the named entities get labeled as "Other" even when one or two labeling functions give the right named entity labels.

5 Summary and Next Steps

In summary, we experiment with different generative models for aggregating labeling function votes for the Named Entity Recognition problem. The results we get so far is not better than those from majority voting. The reason is that with the labeling functions we have so far, there is not much conflict among labeling functions and a generative model is not very helpful in this case. The next step is to design more labeling functions. For example, we are currently exploring the Freebase database and experimenting with how to write additional labeling functions from it.

References

[1] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. Proceedings of the VLDB Endowment 11, 3(2017), 269–282. 2017.

[2] An T Nguyen, Byron C Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. Aggregating and predicting sequence labels from crowd annotations. In Proceedings of the conference. Association for Computational Linguistics. Meeting, page 299. NIH Public Access. 2017.

[3] Neural Architectures for Named Entity Recognition. Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer. NAACL 2016.

[4] Named Entity Recognition with Bidirectional LSTM-CNNs. Jason P.C. Chiu and Eric Nichols. Transactions of the Association for Computational Linguistics (TACL) 4:357-370, 2016.

[5] Hai Leong Chieu and Hwee Tou Ng. Named entityrecognition: a maximum entropy approach using global information. In Proceedings of the 19th internationalconference on Computational linguisticsVolume 1. Association for Computational Linguistics, pages 1–7. 2002.

[6] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the eighteenth international conference on machine learning, ICML. volume 1, pages 282–289. 2001.

[7] Paroma Varma, Bryan He, Dan Iter, Peng Xu, Rose Yu, Christopher De Sa, and Christopher Ré. Socratic Learning: Augmenting Generative Models to Incorporate Latent Subsets in Training Data. arXiv:1610.08123v4 [cs.LG] 28 Sep 2017.

[8] Liyuan Liu, Xiang Ren, Qi Zhu, Shi Zhi, Huan Gui, Heng Ji, and Jiawei Han. Heterogeneous supervision for relation extraction: A representation learning approach. In Proc. of EMNLP. 2017.