Chinese Event Extraction With Inadequate Data

Hanze Dong (15307110244)

***Abstract*—Event extraction is a common application of text mining, which extract main feature of a sentence even a passage. Our experiment mainly focus on the Chinese Event Extraction from independent sentence. Although the techniques to perform a event extraction are relative mature, most of techniques need a large labeled corpus. However, in real problem, it's hard to make a large training data artificially and each kind of training data are indeed specific which may not work in some situations. In our experiment, we test the performance of some Event Extraction Method with about 2,000 sentences in Chinese, and propose some solution to deal with Chinese Event Extraction With Inadequate Data.**

***Index Terms*—Event Extraction, Hidden Markov Chain, Conditional Random Field**

I. Introduction

Event extraction are divided into *Data-Driven Event Extraction* and *Knowledge-Driven Event Extraction*, which results some techniques combining data and knowledge informations.[4] *Knowledge-Driven Event Extraction* can indeed reduce the requirement of large-size data because of the limitation of linguistic patterns, it's difficult to deal with various languages corpus whose grammar are different. Moreover, the idiomatic usage may also be in contradiction with grammar. Considering such problems, *Data-Driven Event Extraction* might be a more flexible choice in Chinese Event Extraction, as the linguistic patterns of Chinese are totally different with English.

Unfortunately, when cannot get a large labeled corpus with any topic, which might make traditional *Data-Driven Event Extraction* not work well with inadequate data: the model are usually over-fitting with such small corpus, even *Naive Bayes* performs the best in some situations.

As a result, we thoroughly evaluate the performance of different types of *Data-Driven Event Extraction* techniques, analyze the advantage and drawbacks of each method, and propose some solutions against the inadequacy of training data.

First, to increase the generalizing ability of our model, it's important to choose a appropriate smoothing method, against the inadequacy of training data.

Second, although it's difficult to get a large labeled corpus, it's possible to get a enormous vocabulary dictionary which can provide us parts of speech (POS), the POS information can also help the performance of our model.

Third, when we deal with the unseen data, interpolation might be useful as a supplement, POS information can play a significant roles in the interpolation.

II. Theory Foundation

A. Hidden Markov Model (HMM)[2]

Assume that O_i and H_i are observation and hidden nodes, h_j and o_j are values sampled from domain of definition $\mathbb{H}$ and $\mathbb{O}$.

Hidden Markov Model is a type of dynamic Bayesian network, which assume the dependency of observation nodes and hidden nodes.

$$P(O_i|\boldsymbol{H}, O_{1:i-1}) = P(O_i|H_i) \tag{1}$$

$$P(H_i|\boldsymbol{O}, H_{1:i-1}) = P(H_i|H_{i-1}) \tag{2}$$

The emission probability $P(o_j|h_i)$ and transition probability $P(h_j|h_i)$ can be estimate from the training data, which construct the whole parameter space.

When we are predicting a hidden chain from observation result, we can use maximum likelihood estimator:

$$\begin{aligned}\boldsymbol{H} := (H_1, ..., H_n) =& \underset{\boldsymbol{H}\in\mathbb{H}^n}{argmax} P(\boldsymbol{O}|\boldsymbol{H}) \\ =& \underset{\boldsymbol{H}\in\mathbb{H}^n}{argmax} \prod_{i=1}^{n} P(H_i|H_{i-1}) \cdot P(O_i|H_i)\end{aligned} \tag{3}$$

$$Note : P(H_1|H_0) = P(H_1)$$

B. Conditional Random Field (CRF)[2]

According to above, Hidden Markov Model mainly focus on the backward hidden nodes and the corresponding observation nodes. However, such dependency assumption is too simple to describe the sentence information, which make us to consider establishing the relationship between hidden chain and observation nodes:

$$P(H_i) = P(H_i|\boldsymbol{O}) \cdot P(H_i|H_{i-1}) \tag{4}$$

From eq.(4), we found that the inference process of parameters are reversed comparing to HMM. In machine learning view, the observation chain and backward nodes become features of hidden nodes, while backward nodes are also unknown. Moreover, because of the probability setting, Conditional Random Field uses log-linear model, so we can get the probability of each hidden node then derive loss function from above:

$$\begin{aligned}P(\boldsymbol{H}|\boldsymbol{O}) =& \frac{exp(\boldsymbol{\omega}^T \Phi(\boldsymbol{H}, \boldsymbol{O}))}{\sum_{\boldsymbol{h_i}\in\mathbb{H}^n} exp(\boldsymbol{\omega}^T \Phi(\boldsymbol{h}_i, \boldsymbol{O}))} \\ :=& \frac{1}{Z(\boldsymbol{H})} exp\{\sum_{i=1}^{n} \boldsymbol{\omega}^T \phi_i(H_i, \boldsymbol{O}, H_{i-1})\}\end{aligned} \tag{5}$$

$$L(\boldsymbol{\omega}) = \sum_{i=1}^{n} \boldsymbol{\omega}^T \phi_i(H_i, \boldsymbol{O}, H_{i-1}) \tag{6}$$

Where Φ, ϕ_i are feature extractors of the model, and $Z(\boldsymbol{H})$ is normalizer.

In such setting, we can learn parameters of $\boldsymbol{\omega}$, CRF++ uses L-BGFS to minimize the loss function, then returns the optimized $\boldsymbol{\omega}$.

C. Smooth and Generalization Ability[3]

As we have inadequate data, the vocabulary set from training data might not cover some word in test data, which may result zero probability in emission matrix and cause some errors (transition of zero makes forward probabilities are all zero and $logP$ overflow). Smooth of our emission matrix is needed for such consideration. Interpolation is a ideal way to smooth the probability distribution, besides the Naive Bayes term, the consideration of POS influences also make sense. Another problem is that the transition matrix is too rugged, such as the transition from 'O' to 'O' is so large that when the information of Event-Vocabulary is not enough, the hidden chain might be all converge to 'O', which raise the idea of punish the probability of such types of transition. According to fig.(1), we can find that the transition diagonal and the transition to 'O' are too high, which may result the converge we talked above.

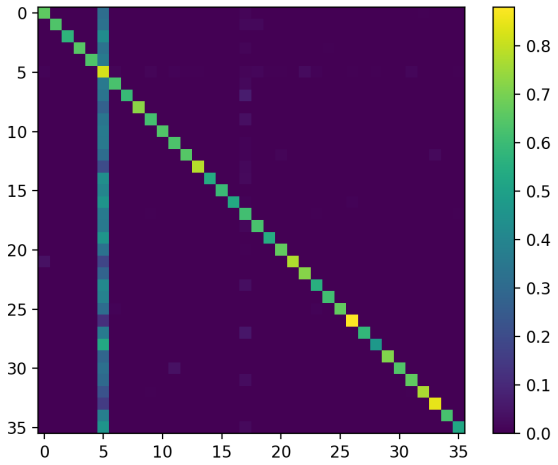

Figure 1. Transition Matrix

Moreover, to avoid zeros in probability matrix, Laplace smoothing seems a reasonable choice, however, the parameters of Laplace smooth should be chosen carefully, which is introduced in the next section.

D. What's more in Event Extraction

1) Triggers: In a simple sentence, there is only one trigger, which can be a limitation for the hidden chain. For example, the probability in HMM can be edit as:

$$P(\boldsymbol{O}|\boldsymbol{H}) = \{\sum_{i=1}^{n} P(h_1|h_1)P(h_1|O_i)\} \times \frac{P(H_k|h_1)P(h_1|H_k)P(H_k|O_k)}{P^2(h_1|h_1)P(h_1|O_k)} \tag{7}$$

$where\ h_1 =' O',\ H_k\ is\ the\ only\ component\ unequal\ to\ 'O'$

2) Arguments: The arguments in a sentence might be more difficult to predict than triggers, for the number of arguments is not fixed,which increase the complexity in our prediction. We choose Viterbi Algorithm in the HMM model, which is widely used in Dynamic Bayes network. Briefly speaking, when the state have Markov Condition, the global shortest path consists a set of local shortest path, when we construct the local shortest path, then we derive the global shortest path by backward calculation, which means we don't need compute all the combination in global view. Due to the limitation of pages, and the detail of Viterbi Algorithm is not what we concerned, we skip this part.

3) Part of Speech (POS) : It's not so difficult to get the POS of a specific word, since there are various dictionaries of any language. So when we have inadequate labeled data for event extraction, it's possible to get information from the dictionary. Due to the limitation of dictionary type we can access, we only tried add POS information in our experiment, which can improve the performance of the model.

III. Result and Analysis

A. Baseline

Although we have several types of method, they are all based on the bayes rule and some conditional assumptions, which means the naive bayes method can be a baseline of any method above, each result poorer than the baseline means over-fitting.

	Tri. (With lim.)*	Tri.	Arg.
Type Corrected	0.9723	0.9051	0.1051
Precision	0.8289	0.2459	0.4022
Recall	0.6861	0.7226	0.9129
F1	0.7508	0.3669	0.5583

Table I
Result of Naive Bayes

*Limitation means the assumption of one trigger of each sentence

B. HMM[2]

When we are use HMM method, it's natural to be cautious about the risk of overfitting. However, although we use add-epsilon to maintain the information in emission matrix, the result is also not satisfying, even worse than the Baseline result. So we firstly, add Naive Bayes interpolation term into the HMM, which can add the generalization ability of our model.

From tab.(II), we found that when we add Naive Bayes interpolation term into the HMM, the result of arguments improve a lot, and the general result is better than either HMM or Naive Bayes, which proves the efficiency of interpolation and HMM indeed works.

Moreover, when we add POS term into the model, the trigger result F1 score can be boosted obviously, which may be result of corresponding between trigger words and verb, also, for arguments, the balance of F1 score type and corrected result can be better with new interpolation. From fig.(2) we can find that the POS type and Trigger is much more related than POS type and argument, which can help to explain the reason of model improvement by POS interpolation. Unfortunately,

the type corrected indeed decrease when we improve the F1 score, for the POS information cannot provide the type, in our setting, when there is only one verb in the sentence, the model will just attach a "T_Movement" which is so ridge. In fact, if we have the dictionary with more information, the situation might be better.

	Original		Interp. N.B.		Interp. POS	
Type	Tri.	Arg.	Tri.	Arg.*	Tri.	Arg.
Type Corr.	**0.97**	**0.43**	**0.97**	0.33/**0.43**	0.89	**0.43**
Precision	0.82	0.76	0.77	0.61/0.72	0.81	0.69
Recall	0.68	0.14	0.74	0.71/0.37	0.79	0.58
F1	0.74	0.24	0.76	**0.66**/0.49	**0.80**	0.63

Table II
RESULT OF HMM

*: Consideration of Type Corrected or F1

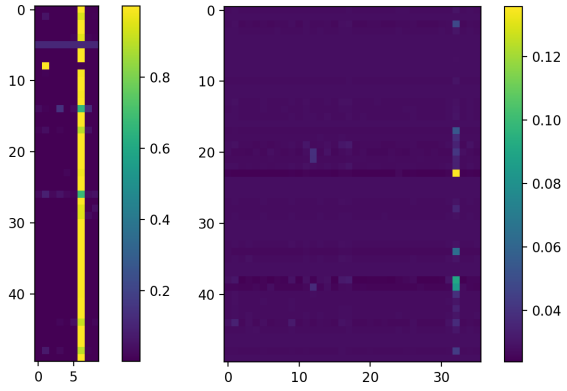

Figure 2. The Relationship between
POS and Trigger (Left) / Argument (Right)

Comment–Why does Laplace Smoothing not work and When does it work: In our experiment, the add-one smooth in emission matrix makes result really bad: the output may converge all 'O' type. As a result, we the addition we use is much less than one (10^{-1} or smaller), that is because of the size of training data is not large enough, the "1" of Laplace Smoothing might be so large, for example, if ('Run','T_Movement') appears only once, when add one, the P('Run','O') can be P('Run','T_Movement')/2, while the transition of ('O','O') is much bigger than the ('O','T_Movement'). As a result, in inadequate data setting, the add-one smooth should be replaced by add-epsilon, which can remain the distribution of original estimation. According to fig.(3), when we use traditional add-k in our smoothing, the distribution become almost flat, that we cannot extract enough information from emission matrix.

	Original		POS Auxiliary	
Type	Tri.	Arg.	Tri.	Arg.
Type Corr.	**0.98**	**0.56**	0.78	0.42
Precision	0.98	0.81	0.92	0.77
Recall	0.36	0.32	0.59	0.59
F1	0.53	0.45	**0.72**	**0.67**

Table III
RESULT OF CRF

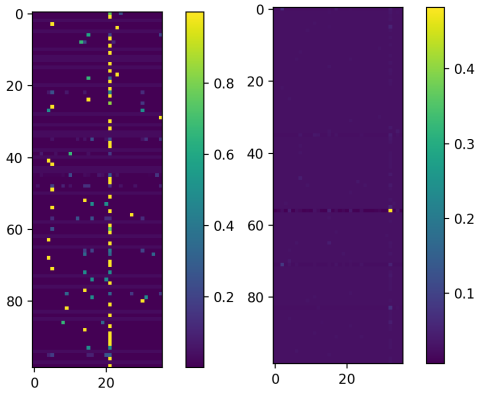

Figure 3. Comparison between Add-epsilon (Left) and Add-K (Right)
(Randomly choose 100 words)

However, when we use POS interpolation, the Laplace smoothing in POS indeed works, that is due to the freedom of POS much smaller than vocabulary's. As a result, before we are using Laplace Smoothing, be attention to the size of our dataset relative to the parameter matrix, which determines the coefficient of add-k smooth.

C. CRF[2][6]

The probability dependency relationship in CRF is much more complex than HMM, which means it has the more potential ability to describe the relationship between hidden chain and observation chain. We use the CRF++[5] in our experiment.

In our experiment, the original CRF tend to overfitting as the inadequacy of training data, however, when we added POS information, the situation ameliorate a lot. The performance of trigger prediction is much better than the original one. As the CRF model doesn't limit the number of trigger appearance, so the performance might be worse than HMM. (P.S. When HMM use Viterbi Algorithm, the F1 Score is also 72%)

About the feature selection of CRF model, we have evaluate the several types of feature selection.

From tab.(IV), we found that the low dimensional feature even performs a good result, which means proves the strong corresponding of connected observation and hidden node, it's similar to MEMM[2], moreover we input the forward information. Comparing to HMM, the HMM hypothesis the hidden nodes are reasons while CRF assume that the observation nodes are reason, but they share the similar information in dataset.

	i		i-1:i+1		i-2:i+2	
Type	Tri.	Arg.	Tri.	Arg.	Tri.	Arg.
Type Corr.	**0.78**	0.42	**0.78**	**0.48**	**0.78**	**0.48**
Precision	0.92	0.77	0.92	0.83	0.92	0.83
Recall	0.59	0.59	0.59	0.54	0.59	0.54
F1	**0.72**	**0.67**	**0.72**	0.65	**0.72**	0.65

Table IV
DIFFERENT FEATURE SELECTION

IV. CONCLUSION AND DISCUSSION

According to our experiments, it's obvious that when we have inadequate data over-fitting occurs usually. Such as HMM and CRF are weaken by the influence with different level. However, besides aware of the over-fitting phenomenon, solutions to alleviates this problem is more important. Thus we propose several approach to deal with the inadequate data problem:

1) Use more information that can be easily accessed: such as POS of words, which can indeed improve the performance, especially for Trigger Prediction
2) Use more limitation in the specific problem: such as the number of trigger in a sentence
3) Use appropriate smoothing method, such as interpolation with low-level feature, and be cautious about the add-k (bad parameters might reduce main information)

1 and 2 suggestions are a low level *Knowledge-Driven Event Extraction* to improve the raw feature expression, while the low level limitation is common in all languages. The third one is a machine learning phenomenon, which should be treated carefully in the specific problem.

REFERENCES

[1] Steven Bird. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.
[2] Michael Collins. Log-linear models, memms, and crfs.
[3] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
[4] Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, and Franciska De Jong. An overview of event extraction from text. In *Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011)*, volume 779, pages 48–57, 2011.
[5] Taku Kudo. Crf++: Yet another crf toolkit. Technical report, 2013.
[6] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.