Chinese Atomic Event Extraction Based on Hybrid Hidden Markov Model

He Zhang

Abstract

The type-dependent event only can express natural language text partially, and the type-independent atomic event is used to express natural language text fully in this paper. A Chinese atomic event extraction method based on Hybrid Hidden Markov Model (HHMM) is proposed via analyzing existing event extraction and Hidden Markov Model (HMM). The HHMM includes a second-order HMM, a two-dimensional HMM, a hybrid Viterbi algorithm and error correction rules. Experimental result shows that this hybrid model can extract atomic events from an unstructured text effectively. The precision, recall and F-score of HHMM are 74.94%, 73.47% and 74.19% respectively.

1 Introduction

The event extraction is the subfield of information extraction, studying how to express natural language text structurally as events, and the event usually holds the structure of “[Who] did [What] to [Whom] [When] and [Where]”. With the development of the Internet, the event extraction has become a research hotspot in Natural Language Processing (NLP). So far, the event extraction has been widely used in many fields, such as textual entailment recognition (Liu et al 2013), social network analysis (Zhou et al 2014), information retrieval (Glavaš et al 2013), stock price prediction (Ding et al 2014), coreference resolution (Lee et al 2012) and community question answering (Nie et al 2013).

Most of current event extraction studies have been focused on evaluation tasks like Message Understanding Conferences, Topic Detection and Tracking, Text Analysis Conference, Automatic Content Extraction (ACE), etc. The traditional event extraction methods usually regard event recognition as a classification problem, mining events from text via machine learning methods or event templates, which can only solve the event extraction problem in certain fields, types or topics. If the machine wants to analyze a paragraph, a discourse and even a document thoroughly, both the major events and affiliated events are needed. For example, the following sentence in Example 1 is selected from the corpus[1] used in ACE event extraction evaluation task, and the type-dependent event in this sentence is expressed as Figure 1.

Example 1：*俄罗斯总统 6 号早上也已经派遣外交部长[伊万诺夫 Artifact] [前往 anchor] [当地 Destination] ，希望协助化解这次南斯拉夫的政治危机。*

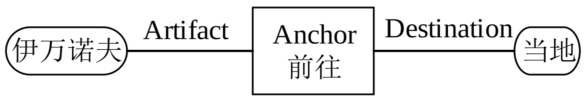

Figure 1. Event for sample sentence

Figure 1 shows that although the ACE has defined eight types and thirty-three subtypes for the concept of event, it is hard to cover all kinds of events in a text. On the contrary, the atomic event can be used to express who, where, when and what in an event more comprehensively in a fine-grained information level. There are six typical kinds of semantic elements in the atomic event, i.e. Agent (A), Patient (P), Predicate (Pred), Time (T) and Location (L). In Figure 2, the sentence in Example 1 is represented as the following atomic event graph.

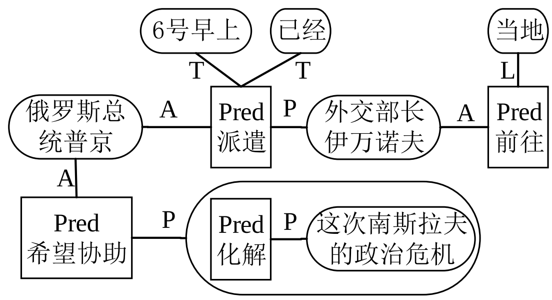

Figure 2. Atomic event graph for sample sentence

To extract atomic events in a text, this paper proposes the HHMM, which takes the sematic features and the relative positions of words into account. Moreover, the error correction rules

[1] http://www.itl.nist.gov/iad/mig/tests/ace/2005/

obtained from the K-means algorithm are used to supplement the statistical model.

The main contributions of this paper are as follows.

(1) Based on the part-of-speech features of Chinese words, the relevance between a state and its historical state has been concerned and a second-order HMM for atomic event extraction is put forward.

(2) With the second-order HMM, the relative positions of atomic event elements has been taken into consideration and a two-dimensional HMM is built to extract atomic events from natural language text.

(3) On the basis of the two-dimensional model, the misclassified atomic event elements are clustered by K-Means algorithm and the error correction rules are concluded which can correct the atomic event extraction result of the statistical model.

The rest of this paper is organized as follows. Section 2 reviews the related word. Section 3 overviews our HHMM. Section 4 then presents experiments and discussions. We conclude our work in Section 5.

2 Related Work

The event extraction can be solved by statistical methods and rules-based methods. Boros et al (2014) approached the task of labeling text spans with event roles by automatically learning relevant features, using a neural model to induce word embeddings in an unsupervised manner and exploiting these word embeddings as features for a supervised event role classifier. The traditional approaches suffer from error propagation since event triggers and arguments are predicted in isolation by independent local classifiers. Li et al (2013) put forward a joint framework based on structured prediction which extracts triggers and arguments together so that the local predictions can be mutually improved. There exists two problems in trigger identification, unknown triggers and word segmentation errors for known triggers. Li et al (2012a) introduced two novel inference mechanisms to explore special characteristics in Chinese via compositional semantics inside Chinese triggers and discourse consistency between Chinese trigger mentions. Llorens et al (2010) analyzed the contribution of semantic roles to TimeML event recognition and classification. For that purpose, an approach using CRF with a variety of morpho-syntactic features and semantic roles features was developed and evaluated. Zhou et al (2014) presented a Bayesian model, called Latent Event Model, to extract structured representation of events from social media. The Chinese event extraction systems suffer much from the low quality of annotated event corpora and the high ration of pseudo trigger mentions to true ones. Li et al (2012b) proposed a joint model of trigger identification and event type determination. To solve the issue that semi-supervised model suffers much from those event mentions, which match infrequent patterns or have no matching pattern, Li et al (2014) proposed various kinds of linguistic knowledge-driven event inference mechanisms for semi-supervised Chinese event extraction. Chen et al (2012) employed a joint modeling approach of event extraction, aiming to address the error propagation problem inherent in Li et al (2012a)'s pipeline system architecture.

HMM (Rabiner L 1989) is a mature and classical statistical model. This model was used widely in recent years. Cahyadi et al (2012) proposed a constrained HMM for aligning the bilingual keyword list of scientific papers to obtain technical terminology dictionaries. Motivated by adaptive rejection sampling and heuristic search, Carter et al (2012) presented a method for extract optimization and sampling from high order HMMs, which is based on sequentially refining a lower-order language model. Engelbrecht et al (2009) introduced a HMMs approach that regard the user's opinion as a continuous process evolving over time to predict judgements about the quality of Spoken Dialog Systems. Ramanath et al (2014) proposed an unsupervised model using HMMs for aligning sections of a collection of roughly similarly-structured legal documents. Garrette et al (2012) improved the MIN-GREEDY algorithm with several intuitive heuristics and demonstrate ways to learn HMM taggers from incomplete tag dictionaries.

At present, most studies are focused on the type-dependent event extraction and the researches on type-independent atomic event extraction are relatively rare. HMM is a mature probability statistical model. The traditional HMM is improved in this paper to solve atomic event extraction. There are three differences between traditional HMM and the HHMM, the first is the second-order HMM, the second is the two-dimensional HMM and the last is the error correction rules.

3 HHMM

In this section, we propose the HHMM to extract type-independent atomic events from Chinese text.

3.1 Model Overview

There are four essential steps included in our model and they are preparatory, model training, classification and error correction. The model overview is shown in Figure 3.

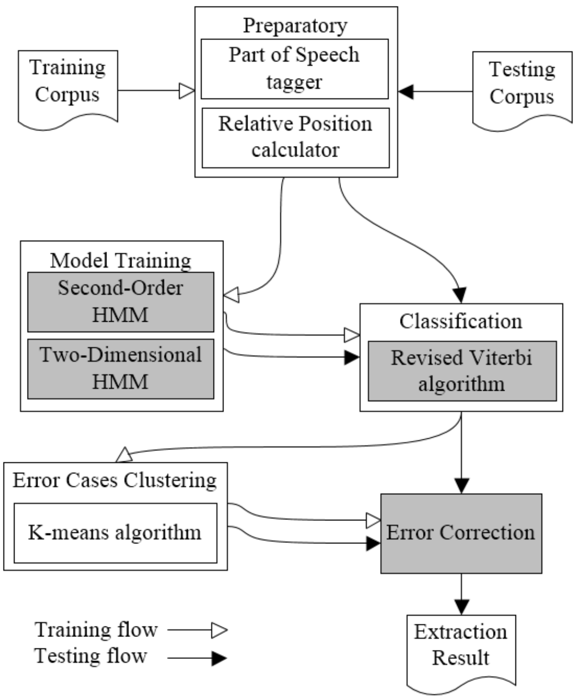

Figure 3. Model Overview

(1) Preparatory

In the preparatory step, all the sentences in the corpus are separated into words using LTP-cloud[2], the corresponding part-of-speech of each word is also annotated during word segmentation and the relative position of each word should be calculated for the following work. As for the training corpus, annotated atomic event tags, part-of-speech tags and relative positions of the words in a sentence are mapped to a three-dimensional vector space. As for the testing corpus, part-of-speech tags and relative positions of the words in a sentence are mapped to a two-dimensional vector space.

(2) Model training

In the training step, both the parameters of the second-order HMM and the two-dimensional HMM should be fitted.

(3) Classification

In the classification stage, the atomic events are extracted from the POS tag vector and the relative position vector of input corpus, by the hybrid Viterbi algorithm based on the model parameters which are obtained from the model training step.

(4) Error Cases Clustering and Error Correction

Chinese is a flexible natural language and the HHMM is a probabilistic graphical model which only considers limited statistical features of words. As supplementary, the linguistic rules, taking more concrete contextual features into consideration, can be used to recognize the atomic events which the statistical model cannot recognize correctly. In this step, some of the misclassifications derived from the classification stage can be corrected.

3.2 Definition of the HHMM

For atomic event extraction, the hidden state sequence is the atomic event tag sequence of a sentence, expressed as $S=\{s_1, s_2, ..., s_i, s_j, ..., s_T\}$, where T is the length of the sequence and s_t denotes the hidden state at time "t" ($1 \le t \le T$). One observation sequence contains POS tags and the other one contains the relative positions. The POS tag sequence is denoted as $PO=\{po_1, po_2, ..., po_i, po_j, ..., po_T\}$, where po_t is the POS tag at time "t"($1 \le t \le T$). The relative position sequence is represented as $RO=\{ro_1, ro_2, ..., ro_i, ro_j, ..., ro_T\}$, where ro_t is the relative position at time "t"($1 \le t \le T$).

The HHMM is defined as $\lambda = (N, M, L, A, B^0, B, C, \pi)$ and the meanings of parameters are listed as follows.

1) N: the number of distinct states in HHMM. For atomic event extraction, N is the number of atomic event element types. The set of different atomic event element types is denoted as $E=\{e_1, e_2, ..., e_N\}$.
2) M and L: the number of distinct observations in HHMM. M is the number of POS tag types and L is the amount of distinct relative positions. The sets of POS tag types and relative position types can be denoted as $P=\{p_1, p_2, ..., p_M\}$ and $R=\{r_1, r_2, ..., r_L\}$ respectively.
3) $A=\{a_{ijk}\}$: the state transition probability matrix. For atomic event extraction, a_{ijk} is the probability that the model will move from the tag pair (e_i, e_j) to the tag e_k.

[2] http://www.ltp-cloud.com/

4) $B^0=\{b_{j(k)}\}$ and $B=\{b_{ij(k)}\}$: the distribution probability matrix of the POS tag observation. The $b_{j(k)}$ is the probability that the k^{th} type of POS tag p_k will be emitted when the model is in state e_j at the beginning of the state sequence. The $b_{ij(k)}$ is the probability that the k^{th} type of POS tag p_k will be emitted when the model stands in state e_j and the previous state is e_i.
5) $C=\{c_{j(k)}\}$: the relative position observation distribution matrix. The $c_{j(k)}$ is the probability that the k^{th} type of relative position r_k will be emitted when the model stands in state e_j.
6) $\pi=\{\pi_i\}$: the initial state distribution. The π_i is the probability that the model will start in state e_i.

The parameters of the second-order HMM and the two-dimensional HMM will be introduced in the following two subsections in detail.

3.2.1 Second-order HMM

The traditional first-order HMM does not consider the effect of contextual features, namely the relevance between the current state and its historical state, which will affect the transition probability and the observation distribution. The second-order HMM can compensate exactly for the above issues. The structures of the second-order HMM is shown in Figure 4.

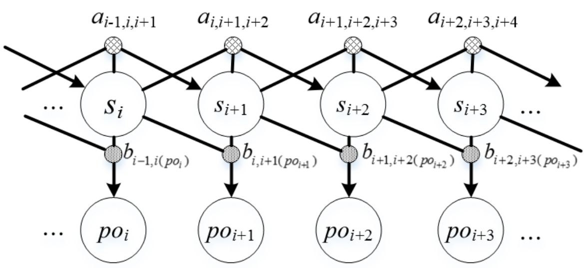

Figure 4 The structure of the second-order HMM

In the second-order HMM, the initial state distribution probability matrix, the state transition probability matrix and the POS tag observation distribution probability matrix are learned from the totally annotated training corpus in a statistical way by the Maximum Likelihood Estimation algorithm (MLE). On the basis of the definition of the HHMM and the procedure of the MLE, the key formulas in the MLE can be derived as follows.

The initial state distribution probability can be calculated by formula (1).

$$\pi_i = \frac{c(e_i)}{\sum_{j\in[0,N]} c(e_j)} \quad (i \in [1,N]) \tag{1}$$

where $c(e_i)$ indicates the sum of the sequences starting from the state e_i in fully annotated training corpus and $\sum_{j\in[0,N]} c(e_j)$ denotes the sum of the sequences starting from any state in fully annotated training corpus.

The state transition probability can be computed by the following formula (2).

$$\begin{aligned} a_{ijk} &= P(s_t = e_k \mid s_{t-1}=e_j, s_{t-2}=e_i) \\ &= \frac{c(e_i,e_j,e_k)}{\sum_{l\in[1,N]} c(e_i,e_j,e_l)} \\ & (i,j,k \in [1,N], t\in[2,T]) \end{aligned} \tag{2}$$

where $c(e_i,e_j,e_k)$ indicates the sum of the times with the state e_i at time "t-2", the state e_j at time "t-1" and the destination state e_k at time "t", and $\sum_{l\in[1,N]} c(e_i,e_j,e_l)$ indicates the sum of the times with the state e_i at time "t-2", the state e_j at time "t-1" and the destination state e_l which might be any type of atomic event tag at time "t".

The following formula (3) and formula (4) can be used to calculate the POS tag observation distribution probability.

$$\begin{aligned} b^0{}_{j(k)} &= P(po_1 = p_k \mid s_1 = e_j) \\ &= \frac{c(e_j,p_k)}{\sum_{l\in[1,M]} c(e_j,p_l)} \\ & (j\in[1,N], k\in[1,M]) \end{aligned} \tag{3}$$

where $c(e_j, p_k)$ denotes the sum of the times with the state e_j at time "1" and the corresponding POS tag observation p_k, and $\sum_{l\in[1,M]} c(e_j,p_l)$ means the sum of the times with the state e_j at time "1" and the corresponding POS tag observation p_l which might be any type of POS tag.

$$\begin{aligned} b_{ij(k)} &= P(po_t = p_k \mid \mathrm{s}_t = e_j, \mathrm{s}_{t-1}=e_i) \\ &= \frac{c(e_i,e_j,p_k)}{\sum_{l\in[1,M]} c(e_i,e_j,p_l)} \\ & (i,j\in[1,N], t\in[2,T], k\in[1,M]) \end{aligned} \tag{4}$$

where $c(e_i,e_j,p_k)$ represents the sum of the times that the state e_i at time "t-1", the state e_j at time "t" and the POS tag observation p_k at time "t", and $\sum_{l\in[1,M]} c(e_i,e_j,p_l)$ means the sum of the times that the state e_i at time "t-1", the state

e_j at time "*t*" and the POS tag observation p_l which might be any type of POS tag at time "*t*".

3.2.2 Two-dimensional HMM

With the observed POS tag, the structure of Chinese sentence and the distribution of different atomic event elements in Chinese sentence are effective to distinct different atomic event elements, composed by the words which have the same or similar POS tags. For example, most of Agent elements and Patient elements are composed by the words who hold the POS tag with noun. It is difficult to deduce which kind of atomic event elements belongs to relying solely on the POS tag observations. However, the relative positions will be helpful because Agent elements usually locate on the front of a sentence and Patient elements usually occur at the behind of an atomic event Predicate. The two-dimensional HMM is shown in Figure 5.

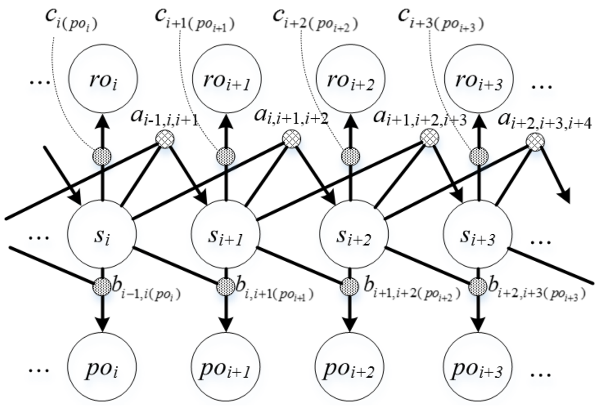

Figure 5. The structure of the two-dimensional HMM

To make use of the two-dimensional HMM, the model parameters should be learned from the training corpus by MLE as same as the second-order HMM. The relative position distribution probability matrix is the only one parameter that need to be learned in this phrase because the other parameters have been built during the estimation of the second-order HMM. Assumed that there are ten relative positions of the words in a sentence, namely from number "1" to "10", the formula (5) can calculate the relative position for a word in a sentence.

$$ro_i = 10 * i / T \quad (i \in [1, T]) \tag{5}$$

The relative position observation distribution probability can be computed by the following formula (6).

$$\begin{aligned} c_{j(k)} &= P(ro_t = r_k \mid s_t = e_j) \\ &= \frac{c(e_j, r_k)}{\sum_{l \in [1,L]} c(e_j, r_l)} \\ &(j \in [1, N], t \in [1, T], k \in [1, L]) \end{aligned} \tag{6}$$

where $c(e_j, r_k)$ indicates the sum of the times with the state e_j at time "*t*", and the corresponding relative position observation r_k and $\sum_{l \in [1,L]} c(e_j, r_l)$ denotes the sum of the times with the state e_j at time "*t*", and the corresponding relative position observation r_l which might be any relative position.

3.3 Hybrid Viterbi algorithm

In our model, given a sequence of POS tag observations $PO=\{po_0, po_1, ..., po_T\}$, a sequence of relative position observations $RO=\{ro_0, ro_1, ..., ro_T\}$ and the HHMM $\lambda = (N, M, L, A, B^0, B, C, \pi)$, we try to find a state sequence S to maximize the probability P($S|PO$, RO, λ). The optimal state sequence S is denoted as $S^*=(s_1^*, s_2^*, ..., s_T^*)$.

There are two Viterbi variables, $\delta_1(i)$ and $\delta_{t,t+1}(j,k)$, and a memory variable $\varphi_{t,t+1}(k)$, need to be defined. The $\delta_1(i)$ is the probability of observation sequences PO and RO with the starting state e_i. The $\delta_{t,t+1}(j,k)$ is the maximal probability of observation sequences PO and RO with the state e_j at time "*t*" and the state e_k at time "*t*+1".The $\varphi_{t,t+1}(k)$ is the optimal state at time "*t*" with the state e_j at time "*t*+1". The formulas (7)-(11) that used in the hybrid Viterbi algorithm are shown as follows.

$$\begin{aligned} \delta_1(i) &= \log(\pi^0{}_i) + \log(b^0{}_{i(po_1)}) \\ &\quad + \mu \log(c_{i(ro_i)}), \quad i \in [1, N] \end{aligned} \tag{7}$$

$$\begin{aligned} \delta_{1,2}(i,j) &= \max_{i,j \in [1,N]} (\delta_1(i) + \log(a_{ij})) + \log(b_{ij(po_2)}) \\ &\quad + \mu \log(c_{j(ro_2)}), i, j \in [1, N] \end{aligned} \tag{8}$$

$$\begin{aligned} \delta_{t,t+1}(j,k) &= \max_{i,j,k \in [1,N], t \in [2,T-1]} (\delta_{t-1,t}(i,j) + \log(a_{ijk})) \\ &\quad + \log(b_{jk(po_{t+1})}) + \mu \log(c_{k(ro_{t+1})}), \\ &\quad j, k \in [1, N], t \in [2, T-1] \end{aligned} \tag{9}$$

$$\begin{aligned} \varphi_{1,2}(j) &= \arg\max_{i \in [1,N]} (\delta_1(i) + \log(a_{ij})) \\ &\quad j \in [1, N] \end{aligned} \tag{10}$$

$$\varphi_{t,t+1}(k) = \arg\max_j(\delta_{t-1,t}(i,j) + \log(a_{ijk})),$$
$$i, j, k \in [1, N], t \in [2, T-1] \quad (11)$$

The μ in above formulas is to control the influence degree of relative position.

Algorithm 1 illustrates how to find the most likely state sequence by the hybrid Viterbi algorithm.

Algorithm 1 the hybrid Viterbi Algorithm

input: POS tag observation sequence *PO*
relative position observation sequence *RO*
HHMM: $\lambda = (N, M, L, A, B^0, B, C, \pi)$
output: the most likely state sequence S^*
//initialization
for *i* **from** 1 **to** *N*
  calculate $\delta_1(i)$ by formula (7);
end for
for *j* **from** 1 **to** *N*
  for *i* **from** 1 **to** *N*
    calculate $\delta_{1,2}(i, j)$ by formula (8);
    calculate $\varphi_{1,2}(j)$ by formula (10);
  end for
end for
//recursion
for *t* **from 2 to** *T*-1
  for *k* **from** 1 **to** *N*
    for *i* **from** 1 **to** *N*
      for *j* **from** 1 **to** *N*
        calculate $\delta_{t,t+1}(j,k)$ by formula (9);
        calculate $\varphi_{t,t+1}(k)$ by formula (11);
      end for
    end for
  end for
end for
//searching for the optimal states sequence
$s_T^* = \arg\max_j[\varphi_{T-1,\mathrm{T}}(j)]$;
for *t* **from** *T*-1 **to** 1
  calculate the t[th] state by $s_t^* = \varphi_{t,t+1}(s_{t+1}^*)$;
end for

3.4 Error correction

Considering the efficiency of statistical model, the number of features and the range of contextual features are limited. Besides, the more features to be taken into account, the more annotated training corpus are needed to fit model parameters. Therefore, it is difficult to distinguish some different types of atomic event elements depending on the small amount of contextual features. This paper makes use of the K-means clustering algorithm to analyze the error cases. At first, the error cases can be classified into six classes depending on their right atomic event element types, and then all error cases of each type will be clustered into three subtypes. The contextual features vector is essential for clustering. For a word in a sentence of training corpus, its contextual features vector of this error case is (e',p,p^1,p^2,p^3,p^4), where e is the annotated tag, e' is the extracted tag, p is the POS tag, p^1 and p^2 are the POS tags of the previous two words, p^3 and p^4 are the POS tags of the following two words.

The K-means algorithm will be executed once for each parent type. For a parent type, three subtype cluster centers can be worked out. While correcting the error recognition for a word, firstly, according to the result of the hybrid model, make sure which parent type does the error most likely belong to, secondly, calculate the distance between the contextual feature vectors of the words and three subclass cluster centers, and finally, correct the error if the distance is less than a threshold T which is calculated during the training procedure. The threshold T for a parent class is calculated by:

$$T = \eta \frac{\sum_{i\in[1,e]} \sum_{j\in[1,k]} |p_i - m_j|^2}{e} + (1-\eta)\frac{\sum_{i\in[1,A-e]} \sum_{j\in[1,k]} |p_i - m_j|^2}{A-e} \quad (12)$$

where A is the number of parent class atomic event elements in training corpus, e is the number of misclassified parent class atomic event elements in statistical result, p_i is the corresponding contextual feature vector, m_j is the j^{th} subclass cluster centers, η is the coefficient to coordinate the influences of two augends.

When the distances between a word and two or more cluster centers less than the corresponding threshold, it will obey the priority sequence of atomic event elements. The priority sequence is: T = L > A = P = Pred > N. For example, according to the computing results, the type of an atomic event element can be both A and L, it will be corrected into L depending on the priority sequence.

4 Experimental result and analysis

4.1 Experimental Background

Although there are some evaluation tasks and evaluation corpus for Chinese event extraction, these resources are concerned about type-related event extraction and the best F-score of the type-dependent Chinese event extraction on ACE corpus until 2014 is 53.9% [Peifeng Li et al., 2012b]. Currently, the research on type-independent Chinese atomic event extraction is rare and there is no a standard corpus for public evaluation. The corpus used in this paper is from the RITE task in NTCIR-9, containing 1414 sentences. We manually annotated all atomic events in the corpus, and the atomic event element distribution is shown in Table 1. The detailed annotating process with human beings is as follows.

(1) All the annotators are divided into 3 groups and annotate all corpus independently, and only discussion within the same group is allowed;

(2) The consistency among three annotated results is checked out. For a sentence, if the three groups' annotated results are consistent, we think the result is right;

(3) For the inconsistent atomic event elements, all the annotators will vote to decide which one is the appropriate annotated result.

Type	Atomic event elements	count
0	A(Agent)	1571
1	Pred(Predicate)	861
2	P(Patient)	2569
3	T(Time)	449
4	L(Location)	443
5	N(Not an atomic event element)	2310

Table 1: Atomic event elements annotated result in NTCIR-9 RITE corpus

To evaluate the performance of the proposed approach, we use *P*(*Precision*), *R*(*Recall*) and *F*(*F*1-*measure*) which are used in the general information extraction systems (Macqueen J 1967). At first, we calculate all *precision*s, *recall*s and *F*1-*measure*s for each type atomic event elements, and then evaluate the overall performance without counting N elements.

4.2 Evaluation result

The overall experiment results of the atomic event extraction from Chinese texts are listed in the following Table 2.

Method	P	R	F
Baseline	54.64%	58.74%	56.61%
+Second-order HMM	58.14%	58.88%	58.51%
+Two-dimensional HMM	57.89%	63.43%	60.53%
+Error Correction	74.94%	73.47%	74.19%

Table 2: Overall performances of all the models

In Table 2, the F-score of the baseline system is 56.61%. The second-order HMM, which considers the relevance between current state and its historical state, makes 1.9% promotion. The two-dimensional HMM, which takes the relative position and the POS tag features into account, gets 3.92% promotion comparing with the baseline model. The error correction rules, which are obtained from the error cases by the K-means algorithm, has 13.66% promotion comparing with the two-dimensional HMM and makes 17.58% promotion comparing with the baseline model.

The performances of the baseline and the second-order HMM are shown in Table 3.

Model	first-order HMM			second-order HMM		
%	P	R	F	P	R	F
A	61.77	57.80	59.72	60.96	63.91	62.40
Pred	54.89	63.18	58.75	59.34	60.16	59.75
P	49.72	64.54	56.17	55.70	59.67	57.62
T	78.30	59.47	67.59	72.11	61.02	66.10
L	36.02	17.16	23.24	41.52	25.96	31.94
N	73.51	58.40	65.09	62.58	60.39	61.47
Overall	54.64	58.74	56.61	58.14	58.88	58.51

Table 3: The performances of the first-order HMM and the second-order HMM

With the historical states in the second-order HMM, the extraction accuracy of the event elements A, Pred, P and L have been made improvement, especially event element L. This phenomenon indicates that it is effective to take the historical states into account in the atomic event extraction. Comparing with the other event elements, the T and N have a slight decline in the extraction accuracy. According to the analysis of the specific extraction data in Table 4, we can drive the conclusion that both the extracted amount and the exactly extracted amount of N and T go up, which makes the precision decline a lot and the recall increase a little.

	T		N	
	HMM1	HMM2	HMM1	HMM2
Annotated	449	449	2310	2310
Extracted	341	380	1835	2229

Exactly Extracted	267	274	1349	1395

Table 4: atomic event elements T and N

As for the two-dimensional HMM, during the training phrase, the coefficient μ is searched from 0 to 1 with the step 0.1, and the performances of models with the different μ are shown in Figure 4.

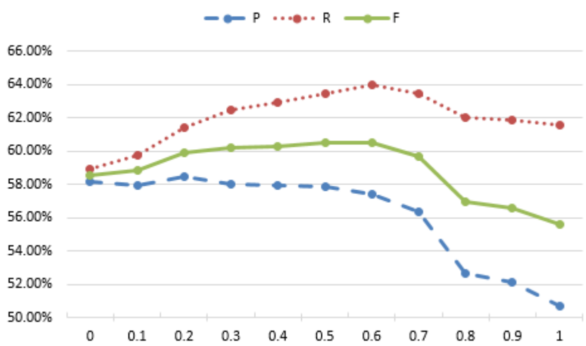

Figure 6. Performances of Two-Dimensional HMMs with different μ

As shown in Figure 6, with the increasing of μ, precision will get lower and lower. When the coefficient is between 0 and 0.6, recall will get higher and higher. However, when the coefficient is bigger than 0.6, the recall is declined. If the coefficient μ is assigned 0.5, the overall performance of model is the best.

The confusion matrix of the results derived from the two-dimensional HMM is shown in Table 5.

	A	Pred	P	T	L	N
A	1099	45	258	16	36	95
Pred	92	500	204	8	11	44
P	300	142	1724	35	68	210
T	56	17	44	280	1	17
L	121	16	142	13	128	13
N	277	133	506	36	36	1300

Table 5: Confusion matrix of Two-Dimensional HMM

In Table 5, it can be found directly that the concrete recognition data for each type atomic event elements. For example, the number 258 at first row and third column indicates there are 258 annotated A in testing corpus was recognized as P in the statistical model. In the process of training, analyzing the error cases by the K-means algorithm to summarize the error correction rules in a statistical way. For each type of atomic events, there are three typical error cases cluster centers. In the process of test, calculate the distances between the contextual feature vector of the words in statistical model result and error case cluster centers by formula 12. As there is a corresponding threshold T for each type atomic event elements, there are 6 thresholds need to be calculated. For each threshold, η coordinates the influences of two augends in formula 15. For each η, the range is from 0 to 1 and the step is 0.1. Table 6 illustrates the confusion matrix after errors correction, and this table shows the best performances of each type atomic event elements with the optimal corresponding η s in the last column.

	A	Pred	P	T	L	N	η
A	1221	48	103	17	39	119	0.6
Tri	53	679	64	8	11	44	0.4
P	74	168	1741	47	98	347	0.4
T	24	17	3	353	1	17	0.6
L	69	16	12	13	310	13	0.5
N	104	147	91	37	41	1868	0.6

Table 6: Confusion matrix after errors correction

Comparing the two confusion matrixes in Table 5 and Table 6, it is obvious that many errors are corrected by error correction rules, especially the most centralized error classes for each type atomic event elements. However, there is still some errors are not corrected, the reason is the error cases are too small that it is not enough to affect the cluster center, as a result, the distances between these errors and the error clusters will be too far.

5 Conclusions and Future Work

In this paper, a hybrid hidden Markov model is put forward for atomic event extraction which can structure the natural language text as atomic event graph more comprehensively than type-dependent event extraction. Compared with the first-order HMM, the second-order HMM takes the transition and observation of current state and its historical state into account. The traditional one-dimensional HMM is expanded to the two-dimensional HMM, both the relative position feature and the POS feature have been taken into consideration in the atomic event extraction. Moreover, as a supplement to statistical model, error correction rules are concluded from error cases and experiment results show that they are effective to improve the performance of our model.

In future work, we will further improve the atomic event extraction statistical model with more rich and effective features. In order to learn the deep sematic of atomic events, we can conduct a cluster analysis on a large number of atomic events and define types on the basis of clustering result.

Acknowledgement

The work presented in this paper is supported by the National Natural Science Foundation of China under Grant No. 61100133 and No. 61173062, the Major Projects of National Social Science Foundation of China under Grant No. 11&ZD189.

References

Maofu Liu, Yan Li and Donghong Ji. 2013. Event semantic feature based Chinese textual entailment recognition. *Journal of Chinese Information Processing*, 2013,05:129-136.

Deyu Zhou, Liangyu Chen and Yulan He. 2014. A simple Bayesian modelling approach to event extraction from Twitter. In *Proc. ACL 2014*, pages 700-705.

Goran Glavaš and Jan Šnajder. Event-centered information retrieval using kernels on event graphs. 2013. In *Proc. of TextGraphs-8 Workshop on EMNLP 2013*, pages 1-5.

Xiao Ding, Yue Zhang, Ting Liu, Junwen Duan. 2014. Using structured events to predict stock price movement: An Empirical Investigation. 2014. In *Proc. EMNLP 2014*, pages 1415-1425.

Heeyoung Lee, Marta Recasens, Angel Chang, et al. 2012. Joint entity and event coreference resolution across documents. 2012. In *Proc. EMNLP 2012*, pages 489-500.

Liqiang Nie, Meng Wang, Gao Yue, Zheng-Jun Zha, Tat-Seng Chua. 2013. Beyond text QA: multimedia answer generation by harvesting web information. In *Multimedia, IEEE Transactions on (Volume:15 , Issue: 2)*, pages 426-441

Qi Li, Heng Ji and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proc. ACL 2013*, pages 73-82.

Peifeng Li, Guodong Zhou, Qiaoming Zhu, et al. 2012a. Employing compositional semantics and discourse consistency in Chinese event extraction. In *Proc. EMNLP 2012*, pages 1006-1016.

Hector Llorens, Estela Saquete and Borja Navarro-Colorado. 2010. TimeML events recognition and classification: learning crf models with semantic roles. In *Proc. COLING 2010*, pages 725-733.

Peifeng Li, Qiaoming Zhu, Hongjun Diao, et al. 2012b. Joint modeling of trigger identification and event type determination in Chinese event extraction. In *Proc. COLING 2012*, pages 1635-1652.

Peifeng Li, Qiaoming Zhu, Guodong Zhou. 2014. Employing event inference to improve semi-supervised Chinese event extraction. In *Proc. COLING 2014*, pages 2129-2139.

Chen Chen, Vincent Ng. 2012. Joint modeling for Chinese event extraction with rich linguistic features. In *Proc. COLING 2012*, pages 529-544.

Denny Cahyadi, Fabien Cromieres, Sadao Kurohashi. 2012. Constrained Hidden Markov Model for Bilingual Keyword Pairs Alignment. In *Proceedings of the 10th Workshop on Asian Language Resources*, pages 85-94.

Simon Carter, Marc Dymetman and Guillanume Bouchard. 2012. Exact Sampling and Decoding in High-Order Hidden Markov Models. In *Proc. EMNLP 2012*, pages 1125-1134.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, et al. 2009. Modeling User Satisfaction with Hidden Markov Models. In *Proc. ACL 2009*, pages 170-177.

Rohan Ramanath, Fei Liu, Norman Sadeh, et al. 2014. Unsupervised Alignment of Privacy Policies using Hidden Markov Models. In *Proc. ACL 2014*, pages 605-610.

Dan Garrette and Jason Baldridge. 2012. Type-Supervised Hidden Markov Models for Part-of-Speech Tagging with Incomplete Tag Dictionaries. In *Proc. CONLL 2012*, pages 821-831.

Rabiner L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. IEEE (Volume:77, Issue:2)*, pages 257-286.

Macqueen J. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281-297.