Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **2**, March **2017**

Design of Chinese Natural Language in Fuzzy Boundary Determination Algorithm Based on Big Data

Fuquan Zhang

School of Software, Beijing Institute of Technology, Beijing, 100081, China
Fujian Provincial Key Laboratory of Information Processing and Intelligent Control
(Minjiang University), Fuzhou 350121, China
Via Room A9#301, Jiyuan, Juyuan, Fuzhou 350008, China
8528750@qq.com

Zijing Mao

Department of Electrical Engineering,
University of Texas at San Antonio One UTSA Circle, San Antonio, Texas, 78249, USA
Via Arrow Oaks 126, San Antonio, Texas, 78249
Zijing.Mao@utsa.edu

Gangyi Ding

School of Software, Beijing Institute of Technology, Beijing, 100081, China
Via Room 611, Software Building, Beijing Institute of Technology,
5 South Zhongguancun Street, Haidian, Beijing, 100081, China
dgy@bit.edu.cn

Lin Xu

Innovative Information Industry Research Institute,
Fujian Normail University, Fuzhou, 350300, China
Via Fuqing Brunch of Fujian Normal University, Campus County 1#
Longjiang Road, Fuqing, Fuzhou, 350300, China
xulin@fjnu.edu.cn

*corresponding author: 8528750@qq.com

Received August, 2016; revised January, 2017

Abstract. *In this study, we proposed a new method to determine the fuzzy boundary of natural language based on big data. According to the principle of natural language recognition, the acoustic characteristics of the analysis, natural language acoustic model and statistical model is established, Viterbi decoding algorithm has been applied for natural language decoding, and take this as the basis, we implemented deep learning methods for natural language feature extraction, and used the support vector machine method for classification, according to the characteristics of natural language significant division, in accordance with the constraints, combined with large data analysis method, to determine the natural language fuzzy boundary. Experimental results indicate that using the proposed method, compared with existing methods the recall and accuracy rate have been improved.*

Keywords: Natural language processing, Fuzzy boundary, Determination method, Big data

---

1. **Introduction.** Natural language usually refers a language that will evolve with the culture; it is created for a specific purpose and has a certain cognitive nature and uncertainty [1, 2]. With the development of computer technology, using natural language to communicate with the computer is the long-term goal for researchers [3, 4]. In that way, people can use their most used language to use a computer, and does not need to spend a lot of time and effort to learn a variety of computer languages [5, 6]. It can also learn more about human language ability and intelligence mechanism. Therefore, natural language research is the foundation for effective communication between man and computer, computer and artificial intelligence and is an important direction [7]. However, many fuzzy boundaries [4, 5] cannot be well defined based on various application fields, result in the low efficiency of identify natural language features and this study aims at handling the fuzzy boundary problem.

Many works have also been proposed in the past decades. Among these works, [6] proposed a word connection method to define the fuzzy boundary based on natural label resources, this method mainly build a dynamic corpus from the natural language environment and based on the corpus to build word connections instances to extract natural language features, which are then applied to classify the fuzzy boundary; [7] tries to establish language entity relationship model and deduced the structure of natural language according to the process model and algorithms that specify linguistic analysis; [8] aims at solving the presence of coarse-grained natural language words cannot express the complexity of the relationship between objects. It defines the boundary based on cognitive linguistics theory, intention maps were showing the contact among object nodes are applied in this method, natural language attribute space have been build and conversion process for the operation series have been revised. Although this method can define boundaries well for concrete natural language, the boundary for abstract natural language still cannot be well defined. To solve these problems, we proposed a new method for natural language fuzzy boundary determination. We apply acoustic features analysis method to build natural language acoustic model and use Viterbi decoding algorithm to decode the natural language. Then deep learning methods are applied to identify and extract natural language features. Support vector machine (SVM) [9] is used to classify significant discriminated features and then fuzzy boundaries is defined. Our experiment results have shown the advantage of applying this method comparing to existing algorithms in improving both precision and recall rate.

2. Natural Language Decoding Methods.

2.1. Natural language processing methods.

The goal of natural language processing is to convert the language signal to the corresponding text information. Such system mainly includes natural language extraction, linguistic model, acoustic model and decoder. The training process for natural language is first obtain the acoustic features extracted from the original waveform language data trained to give the acoustic model, and voice dictionary, language model to form a network of new language feature extraction via acoustic model representation, recognition result by the Viterbi decoding [10].

One commonly applied model is Hidden Markov Model (HMM) [11] for the training process. That is, given a series of natural language signal $\mathrm{O}_1^{\mathrm{T}} = \{\mathrm{O}_1, \mathrm{O}_2, \ldots, \mathrm{O}_{\mathrm{T}}\}$, we have the maximum posterior probability for the output text sequence, expressed as:

$$\tilde{W} = \arg\max_{W} P\left(W \left| O_1^T \right.\right) = \arg\max_{W} \frac{P\left(O_1^T \left| W \right.\right) P\left(W\right)}{P\left(O_1^T\right)} \tag{1}$$

where $\tilde{W}$ is the output text sequence. P(W) is the prior for natural language model. $P(O_1^T|W)$ is the acoustic model, represents given a text sequence, the probability of O_1^T as the acoustic features. $P(O_1^T)$ is the probability of observing acoustic features, which is independent from W and thus we can convert (1) to (2).

$$\tilde{W} = \arg\max_{W} P\left(O_1^T \left| W\right.\right) P\left(W\right) \tag{2}$$

2.2. **Acoustic feature analysis.** Acoustic features are extracted from acoustic signals and influenced much on the natural language recognition. Therefore, how to extract discriminate features required to be investigated, especially to extract those features from people with different area, age and accents. The feature extraction process can be regarded as the signal compression process with the object to achieve the best classification results. We have observed acoustic signal to be stable in the range of 10 to 30 ms, then we can apply short time analysis on acoustic signals. Such feature includes linear predictive coefficients, Cepstral Coefficients, Mel-Frequency Cepstral Coefficients (MFCC), perceptual linear predictive coefficients (PLP) and so on, among which cepstral coefficients is one of the most important feature for natural language and the formula as follows:

$$CEP\left(t\right) = DEF^{-1}\left(\ln\left|DEF\left(Frame\left(t\right)\right)\right|\right) \tag{3}$$

where Frame(t) is the language signal for t^{th} frame. $\mathrm{DFT}\left(\bullet\right)$ and $\mathrm{DFT}^{-1}\left(\bullet\right)$ represents discrete Fourier transform and inverse discrete Fourier transform. In order to improve the robustness of the feature and reduce the feature dimension, we applied normalization after feature extraction:

$$P\left(s\left|w_i'\right.\right) = \frac{N\left(s, w_i'\right)}{N\left(w_i'\right)} \tag{4}$$

where $\mathrm{N}\left(\mathrm{s}, \mathrm{w}_{\mathrm{i}}'\right)$ denotes the appearance times of acoustic signal features s and training corpus w' pairs. $N(w')$ is the appearance times of corpus w'.

2.3. **Build acoustic model.** Acoustic model plays an important role in the natural linguistics, represents a shift in the process of generating an acoustic signature sequence of primitives [12]. Given an acoustic feature sequence, the maximum probable state sequence corresponding to the feature sequence can be calculated based on the acoustic model with maximum likelihood method. One of the most commonly used acoustic model is hidden Markov model (HMM), which can use hidden states to describe time serial signals. HMM, which has prominent capability in modeling dynamic time series, can be divided into two parts: the hidden states and observable units. The probability of a hidden state can be calculated by the corresponding observable unit at current time and hidden state of previous time. HMM can be described with following 5 parameters.

$$M = \{S, O, A, B, \pi\} \tag{5}$$

where S is the hidden state set, O is the observable set, A is the transition probability set, B is the emission probability set, and π is the initial probability set. The natural language acoustic model is shown in figure 1, where a_{ij} is the transition probability from i to j.

The emission probability can be modeled by mixture Gaussian distribution to fit the acoustic features and denoted as B:

$$b_{ij}\left(O\right) = P\left(O\left|i,j\right.\right) = \frac{1}{\left(2\pi\right)^{p/2}\left|\sum_{ij}\right|^{1/2}} * \exp\left\{-\frac{1}{2}\left(O-\mu_{ij}\right)\sum\nolimits_{ij}^{-1}\left(O-\mu_{ij}\right)\right\} \tag{6}$$

where i,j denotes as two neighbor time states, P is the mixture number, μ_{ij} and Σ_{ij} represents the mean and standard deviation from states at time i to j.

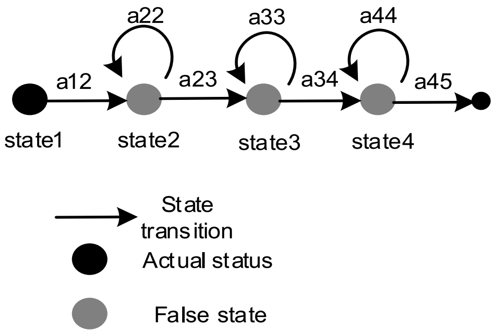

FIGURE 1. HMM acoustic model for natural languag

2.4. **Build linguistic model.** Based on acoustic model we build linguistic model, which mainly models the probability of a word sequence shown in the natural language environment. Assume the word sequence is $\mathrm{W}_1^{\mathrm{n}} = \{\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_{\mathrm{n}}\}$, and the probability of the sequence is:

$$P\left(W_1^n\right) = P\left(W_1\right) P\left(W_2 \left| W_1\right.\right) P\left(W_3 \left| W_1 W_2\right.\right) ... P\left(W_n \left| W_1 W_2 ... W_{n-1}\right.\right) \tag{7}$$

where $\mathrm{P}(\mathrm{W}_1)$ is the W_1 appearance probability, $\mathrm{P}(\mathrm{W}_2|\mathrm{W}_1)$ is the conditional probability which can be used calculate the probability of n^{th} word based on previous n-1 word. Then, we have the natural linguistic model:

$$P\left(W_1^n\right) = P\left(W_1\right) P\left(W_2 \left| W_1\right.\right) P\left(W_3 \left| W_1 W_2\right.\right) ... P\left(W_n \left| W_1 W_2 ... W_{n-1}\right.\right) \tag{8}$$

2.5. **Viterbi decoding analysis.** We applied Viterbi decoding algorithm after training of acoustic HMM. Viterbi algorithm will provide a best state path based on dynamic programming [12]. We also introduced weighted finite state transducer (WFST) into speech recognition to construct large scale static networks. Such network includes many types of natural language prior knowledge, such as voice dictionary, language model, context dependency and HMM. The decoding algorithm of WFST is as follows:

$$f\left(o, h, c, l, \omega\right) = \sum_{l \in L} \sum_{c \in C} \sum_{h \in H} f\left(o \left| h\right.\right) f\left(h \left| c\right.\right) f\left(c \left| l\right.\right) f\left(l \left| \omega\right.\right) f\left(\omega\right) \tag{9}$$

where o is the natural language feature sequence, ω is the optimal word sequence, $f(\omega)$ is the natural language model, $f(l|\omega)$ is the voice dictionary, $f(c|l)$ is the context dependent model, $f(h|c)$ is the HMM, $f(l|\omega)f(\omega)$ is the phoneme constructed network structure, $f(c|l)f(l|\omega)f(\omega)$ is the context dependent phoneme constructed network (C-level network), $f(h|c)f(c|l)f(l|\omega)f(\omega)$ is the HMM structured network (H-level network). Based on the WFST theory, we can combine different type of knowledge and optimize both C-level and H-level network:

$$Network_{C-level} = C \circ \min\left(\det\left(L \circ G\right)\right)$$

$$Network_{H-level} = \min\left(\det\left(H \circ \det\left(C \circ \det\left(L \circ G\right)\right)\right)\right) \tag{10}$$

where $^\circ$ is the combined operation, $\det(\bullet)$ is the determination operation, $\min(\bullet)$is the minimization operation. The static search network is shown in figure 2.

Assume we have $\mathrm{v} = [\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_{\mathrm{B}}]$, v_{B} is feature value at the B^{th} frequency band. Then the activated units can be defined as:

$$h_{j,k} = \theta\left(\sum_{b=1}^{s} W_{b,j}^T v_{b+k-1} + a_j\right) \tag{11}$$

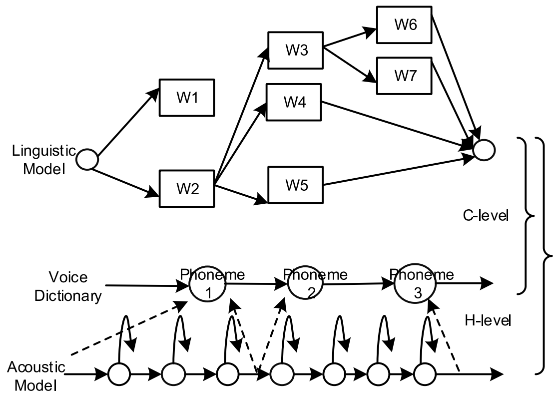

FIGURE 2. C-level and H-level networks in natural language recognition

where $h_{j,k}$ is the output layer of the j^{th} feature maps corresponding to the k^{th}activated units, s is the convolutional kernel length, $W_{b,j}^{T}$ is weight for the j^{th} output units at b^{th} frequency band at time T, a_j is the bias for j^{th} feature map,and $\theta(\bullet)$ is the activation function. Normally sub-sampling is applied to reduce the feature size and we applied max-pooling here:

$$P_{j,m} = \max_{k=1}^{r} \left(h_{j,(m-1)\times n+k}\right) \tag{12}$$

where $P_{j,m}$ is the output for natural language, j is the feature map index for input layer, m is the feature map index for output layer, n is the sub-sampling factor, r is the decoding size, represents the unit size for decoding.

3. **Deep learning based natural language speech recognition.** In 2006, deep belief network [13], which further developed as deep neural network (DNN) [14], has been proposed by Geoffrey Hinton. Within one decade, many other deep learning structures have been proposed such as deep auto-encoder (AE) [15], convolutional neural network (CNN) [16] and LSTM recurrent neural network (RNN) [17]. Applications based on deep learning, such as GoogLeNet [18] and VGGNet [19]constructed by deeper CNNs, deep face [20] and Memory Net [21] in Facebook and so on have been shot out especially in recent few years, for the capability of analyzing data using deep structure on big data. DL beats most state-of-arts machine learning algorithm in different areas including speech recognition [22]. The training process is as follows:

3.1. **Natural language pre-training.** Restricted Boltzmann machine (RBM) [23] has been applied for the pre-training process. The RBM is an important component for both DNN and DSN. RBM is a type of undirected graphical model consisting of a layer of visible units (input data) and a layer of hidden units. In this work, we consider the Gaussian-Bernoulli RBM .[22], where hidden units have binary values but the visible units have real values. Let $h \in \{0,1\}^{N\times 1}$ represent a vector of N hidden units, respectively. Their connections are defined according to an energy function:

$$E(v,\ h) = -b^{T}v - c^{T}h - v^{T}Wh \tag{13}$$

where W is an M×N matrix that defines the weights between each pair of visible and hidden units, and b and c are M×1 and N×1 bias vectors for visible and hidden units, respectively. Based on the energy function (2), the joint distribution of v and h is:

$$P(v,\ h) = \frac{e^{-E(v,\ h)}}{Z} \tag{14}$$

where Z is a normalizing constant. Training of RBMs involves estimating the parameters $\theta = (W, b, c)$ from (14) given visible units based on the maximum likelihood criterion, which is analytically intractable. A stochastic optimization scheme based on contrastive divergence was proposed in [24] to obtain a computationally efficient but suboptimal solution. The stochastic search includes a free parameter called the learning rate that users must either predefine or determine empirically. Moreover, in the stochastic search, the conditional activation probabilities of individual hidden units are given by:

$$\mathrm{p}\left(\mathrm{h_n}=1|\mathrm{v}\right)=\sigma\left(\sum_{\mathrm{m=1}}^{\mathrm{M}}\mathrm{w_{mn}v_m}+\mathrm{c_n}\right),\forall\mathrm{n} \tag{15}$$

and the conditional probabilities of individual visible units are approximated by:

$$\mathrm{p}\left(\mathrm{v_m}|\mathrm{h}\right)=\mathrm{N}\left(\sum_{\mathrm{n=1}}^{\mathrm{N}}\mathrm{w_{mn}h_n}+\mathrm{b_m},\ 1\right),\forall\mathrm{m} \tag{16}$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathrm{N}(\cdot,\cdot)$ is the Gaussian distribution with mean $\sum\limits_{\mathrm{n=1}}^{\mathrm{N}}\mathrm{w_{mn}h_n}+\mathrm{b_m}$ and variance 1. The conditional distribution (16) can be used to simulate the pattern of the visible units for an activated hidden unit. Particularly, when setting $\mathrm{h_i}=1$ and $\mathrm{h_j}=0\ \forall\ \mathrm{j}\neq\mathrm{i}$, the distribution (16) becomes:

$$\mathrm{p}\left(\mathrm{v_m}|\mathrm{h_i}=1,\mathrm{h_j}=0\ \forall\ \mathrm{j}\neq\mathrm{i}\ \right)=\mathrm{N}\left(\mathrm{w_{mi}}+\mathrm{b_m},\ 1\right),\forall\mathrm{m} \tag{17}$$

From (17), we can see that the $\mathrm{w_{mi}}$ plus the unit biases $\mathrm{b_m}$ define the activation patterns of the visible units. We can also sample out new visible units based on (17) as $\mathrm{v_m}'$, nd based on (15) to sample new hidden units using $\mathrm{v_m}'$ and denotes as $\mathrm{h_n}'$. When applied the contrast divergence [25] algorithm, we can update weights based on the following equation:

$$\Delta w_{mn}=\varepsilon\left(\langle v_m h_n\rangle-\langle v'_m h'_n\rangle\right) \tag{18}$$

where $\langle\cdot\rangle$ denotes the expectation of samples, ε is the learning rate.

3.2. **Fine-tuning steps.** After applied RBM with unsupervised learning, we used back-propagation to fine-tuning the parameters in the deep learning networks. Assume we have M samples denote as $\left\{\left(\mathrm{x}^1,\ \mathrm{y}^1\right),\ \ldots,\left(\mathrm{x}^\mathrm{M},\ \mathrm{y}^\mathrm{M}\right)\right\}$. and apply gradient decent algorithm to calculate weights. The loss function is as follows:

$$J\left(W,b;x,y\right)=\frac{1}{2}\left\|h_{W,b}\left(x\right)-y\right\|^2 \tag{19}$$

Assume we have M samples $\{(\mathrm{x}^1,\ \mathrm{y}^1),\ \ldots,(\mathrm{x}^\mathrm{M},\ \mathrm{y}^\mathrm{M})\}$, where $\mathrm{y^i}\in\{1,2,\ldots,\mathrm{k}\}$, our classification goal is to calculate the probability of $p(y=j|x)$. Therefore, the output units should be:

$$h_\theta\left(x^{(i)}\right)=\frac{1}{\sum\limits_{j=1}^{k}e\theta_j^T x^{(i)}}\begin{bmatrix} p\left(y^{(i)}=1\left|x^{(i)};\theta\right.\right)\\ p\left(y^{(i)}=2\left|x^{(i)};\theta\right.\right)\\ \ldots\\ p\left(y^{(i)}=k\left|x^{(i)};\theta\right.\right)\end{bmatrix} \tag{20}$$

where $\theta\in\mathrm{R^{n+1}}$ as the natural language model parameters, $\frac{1}{\Sigma_{\mathrm{j=1}}^{\mathrm{k}}\mathrm{e}\theta_\mathrm{j}^\mathrm{T}\mathrm{x^i}}$ is the normalization process for the output probability distribution. Then the output units at layer l can be calculated based the the weights and units from layer l-1:

$$u_i=\sum\nolimits_{j=0}^{j<N(l-1)}w_{ij}\cdot x_j+\theta_i \tag{21}$$

where N(l-1) is the number of hidden unit size at layer l-1, x_j is the j^{th} unit at layer l-1, w_{ji} is the weights connected between unit j at layer l-1 and unit i at layer l and θ_i is the bias. Assume we have m input unit and n hidden units, we have the energy function (13) rewrite as:

$$E\left(u,h\left|\theta\right.\right)=-\sum_{i=1}^{m}a_iu_i-\sum_{j=1}^{n}b_jh_j-\sum_{i=1}^{m}\sum_{j=1}^{n}u_iW_{ij}h_j \tag{22}$$

where $\theta=\{\mathrm{w_{ij}},\mathrm{a_i},\ \mathrm{b_j}\}$ as model parameters.

4. **Natural language fuzzy boundary determination algorithm.** After applied deep learning on natural language speech recognition, we applied support vector machine (SVM) for determine natural language fuzzy boundary. Given a sample $\mathrm{D_i}=(\mathrm{x_i},\mathrm{y_j}),\mathrm{i}=1,\ldots,\mathrm{l},\mathrm{y_i}\in\{1,-1\}$ from training dataset D, where x_i is the input data, l is the total sample size, y_i indicates the two types of natural language. In order to separate out the given two types, max-margin is applied as the objective function:

$$\min\frac{1}{2}\left\|w\right\|^2$$

$$subjec\ to\ yi\left[\left(WX_i\right)+b\right]\geq1-\varepsilon_i,i=1,2,\ldots,l \tag{23}$$

When the training set is linear and cannot be divided, we need to introduce new variable which is slack variable $\varepsilon_i>=0$, we have the expression converted to:

$$\min\frac{1}{2}\left\|w\right\|^2+c\sum_{i=1}^{l}\varepsilon_i$$

$$subjec\ to\ yi\left[\left(WX_i\right)+b\right]\geq1-\varepsilon_i,i=1,2,\ldots,l \tag{24}$$

where c is the penalty factor, normally greater than 0, indicating the penalty degree for classifying a wrong sample, $\varepsilon_i>=0$. We then introduced Lagrange function:

$$L=\frac{1}{2}\left\|w\right\|^2-\sum_{i=1}^{l}\alpha_iy_i\left[\left(wx_i\right)+b\right]+\sum_{i=1}^{l}\alpha_i \tag{25}$$

where $\alpha_i>0$ as the Lagrange multiplier, and the optimal classification function can be calculated as:

$$f\left(x\right)=sgn\left\{\left[\sum_{j=1}^{l}\alpha_j^*y_j\left(x_i\cdot x_j\right)\right]+b^*\right\},x\in R^n \tag{26}$$

And the following constraint should be satisfied:

$$\iint K\left(x_1,x_2\right)f\left(x_1\right)f\left(x_2\right)dx_1dx_2\geq0 \tag{27}$$

Then $\mathrm{d}\left(\mathrm{x_1}\right)<\mathrm{d}\left(\mathrm{x_2}\right)$ indicates when satisfied (27), the fuzzy boundary can be determined.Assume m_n is the natural language attribute expectation when classification and m_f is the expectation as input, then we have the optimal threshold:

$$V_{th}=0.5\left(m_n+m_f\right) \tag{28}$$

We separate natural language into two types: ascending and descending attributes, where ascending indicates the natural language has high attribute value while descending has low attribute value. Assume we have ascending natural language attribute, then the determination formula is as follows:

$$B = \begin{cases} 1 & V \geq V_{th} \\ 0 & V < V_{th} \end{cases} \tag{29}$$

And descending determination is the opposite:

$$B = \begin{cases} 0 & V > V_{th} \\ 1 & V \leq V_{th} \end{cases} \tag{30}$$

where B is the determination results, 1 indicates judgement decision, 0 indicates no judgement decision and V is the natural language data perception reading. Therefore, we have convert the fuzzy boundary determination to detection of contour for threshold V_{th}, if the natural language data is closer to the contour for V_{th}, the corresponding perception attribute reading. So we can apply unstable sort for the neighbor data reading based on natural language perception readings values, for ascending natural language attribute, we applied descending order, while for descending natural language attribute, ascending order is applied. Then the location of V_{th} in the attribute sequence can be detected and both left and right N/2 attribute values are selected based on the location. For multiple attribute natural language fuzzy boundary determination, assume m_n^i is the attribute reading expectation,m_f^i is the attribute determination expectation, we have the optimal threshold for each attribute i:

$$V_{th}^i = 0.5\left(m_n^i + m_f^i\right) \tag{31}$$

where V^i is the reading for attribute i. After the determination of boundary for each attribute, we have the multi-attribute natural language boundary determination as:

$$B = \bigcap_{i}^{M} B_i \tag{32}$$

where B_i is the determination result for i^{th} attribute and the fuzzy boundary determination should satisfy the closed curve formed by $V^1 = V_t h^1, V^2 = V_t h^2, ..., V^M = V_t h^M$, where M is the total amount of attribute. Then the fuzzy boundary can be determined by fitting the closed curve. Assume natural language attribute S has N maximum attribute value across X cordinate, denotes as $N_{X\max}$ we have the coordinate denotes as $(X_{\max}, Y)$, and for the minimum attribute value, denotes as $N_{X\min}$ we have the coordinate denotes as $(X_{\min}, Y)$, similarly for coordinate Y we also have $Y_{\max}$ and $Y_{\min}$,then the fitting function can be expressed as:

$$\begin{cases} y = ax^2 + bx + c & X_{\max} - X_{\min} \geq Y_{\max} - Y_{\min} \\ x = ay^2 + by + c & X_{\max} - X_{\min} \prec Y_{\max} - Y_{\min} \end{cases} \tag{33}$$

If $X_{\max} - X_{\min} \geq Y_{\max} - Y_{\min}$, $y = ax^2 + bs + c$ will be used for fitting the curve, otherwise we applied $x = ay^2 + by + c$ for curve fitting. Assume the square distance of attribute 1 to attribute 2 is denoted as $y_i = p(x_i)$, we have the polynomial curve fitting function as:

$$I = \sum_{i=0}^{m} \left[p\left(x_i\right) - y_i\right]^2 \tag{34}$$

We can calculate the parameters for the curve fitting as:

$$\frac{\partial I}{\partial a_j} = 2\sum_{i=0}^{m}\left(\sum_{k=0}^{n} a_k x_i^k - y_i\right) x_i^j = 0 \qquad j = 0, 1, \ldots, n \tag{35}$$

We also define the threshold for fuzzy boundary determination:

$$\begin{cases} 1 & D \leq D_t \\ 0 & D > D_t \end{cases} \tag{36}$$

where D_t is the defined threshold. When $D \leq D_t$, we have D=1 indicates the data is not the margin while $D > D_t$, the natural language data is close to the fuzzy margin and is thus defined as the required margin.

5. **Simulation and Results.**

5.1. **Experiment data.** In this experiment, we used 863 Chinese speech database as experiment data, which includes 166 subjects with half female and half male. The reading voice library is selected from People's Daily in 2014 to 2015 in total 1560 paragraph and includes 42638 sentences. The sampling rate is 16k Hz. We separate data into training and testing data where we randomly picked 2000 voice sentence as testing and remaining as training data. Our model is trained with 3.2 GHz CPU, GTX 660ti GPU. We randomly picked 5% from each subject in the training data as validation set. Normalization is first calculated for each individual on validation set to have zero mean and unit variance and applied over the entire corpus to avoid re-estimation of distribution for training set.

5.2. **Experiment process.**

1. Feature extraction

   Each speech resource was analyzed using a 25-ms Hamming window with 10-ms between the left edges of successive frames. In all the experiments, we represented the speech using 12th-order MFCC and energy, along with their first and second temporal derivatives. In total we have ~10 million training samples.
2. Voice attribute extraction

   We applied attribute extraction based on deep neural networks for each voice attribute. The trained attribute extractor is used to construct the required attributes, according to the knowledge of the phonetics speech attribute into 21 categories, and the deep neural network architecture is 392567207202562,where 39 is the network training input, 256720720256 are the size of the successive 4 hidden layers, and 2 represents the label layer with two classes (ascending/descending attributes). We combined 2 label units for each of 21 attributes together and come out with a 212=42 dimension feature vector as the extracted attribute for fuzzy boundary determination based on SVM.
3. Natural language fuzzy boundary determination

   Based on the deep learning extracted features we applied SVM to classify the natural language features and determine the fuzzy boundary.

5.3. **Experiment result.** We calculate both precision and recall of fuzzy boundary determine. Assume the natural language total feature size is N with set B, the SVM classifier returned fuzzy boundary determination size is N_R,and the actual determined fuzzy boundary size is N_A with set A, we have the precision and recall as follows:

$$\Pr ecision = \frac{P(A \cap B)}{P(B)} = \frac{N_R}{N} \tag{37}$$

$$Recall = \frac{P(A \cap B)}{P(A)} = \frac{N_R}{N_A} \tag{38}$$

We applied our proposed method and two other determination methods from [7] and [8] for comparison the time costs are shown in figure 3. We can see the average time costs for [7] is about 6s, and its determination time increases when the fuzzy boundary size increases; [8] has shorter determination time (∼4.2s) than [7], but the time increasing slope is still large as fuzzy boundary size increases; while our proposed method only have determination time at ∼2.1s and the time increasing slope is the smallest among three algorithms.

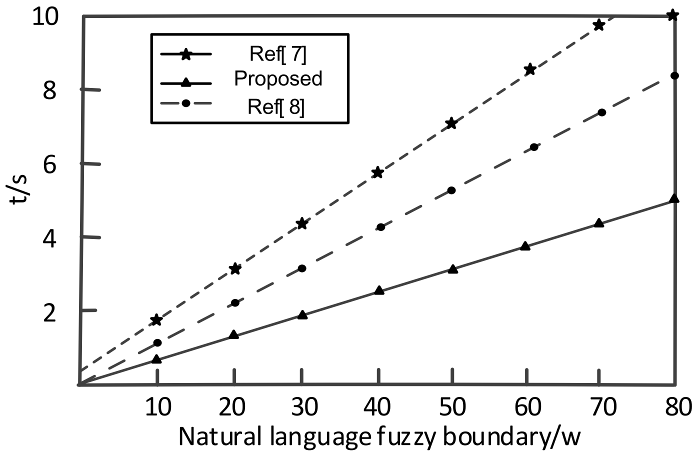

Figure 3. Time costs with three different algorithms

Table 1 also shows the determination precision for the given three algorithms. We can see the average accuracy for [7] is at 61.25%, and as time increases the accuracy increasing speed is slow; for [8] the average accuracy is at 66.25%, and as time increases the determination accuracy also exhibits some volatility, indicating low robustness; while our proposed method has achieved average accuracy at 86.88%, and the improvement as time increases is also stable, comparing with [7] and [8], the average improvement for our proposed algorithm are 25.63% and 20.63%.

Table 1. Determination precision of three different algorithms

Time(s)	Ref [8](%)	Proposed(%)	Ref [7](%)
50	43	56	22
100	56	79	34
150	59	84	45
200	75	85	47
250	68	94	76
300	63	97	85
350	80	100	89
400	86	100	92

Table 2 depicts the determination recall for the given three algorithms. We can see both [7] and [8] are suffering fluctuation of recall as time increases and for [7], the precision is ∼25% on average; [8] has smaller precision at ∼10% on average and it could not increase the recall rate after 80s; while our proposed method, although have fluctuation on recall rate, the average is at ∼41%, which is 16% and 31% over the methods [7] and [8].

6. **Conclusion.** In this paper, we proposed a new method for natural language fuzzy boundary determination. The method first extracted linguistic features and build an acoustic model based on deep neural network and HMM model. Support vector machine has been applied on the feature extracted from deep learning to determine the fuzzy

TABLE 2. Determination recall of three different algorithms

Time(s)	Ref [8](%)	Proposed(%)	Ref [7](%)
10	20	50	33
20	18	48	35
30	21	40	30
40	24	51	32
50	20	48	28
60	18	45	30
70	21	46	34
80	19	50	31
90	15	40	25
100	14	45	30
110	10	36	28
120	9	40	26

boundary. Our experiment indicates the proposed methods can improve both precision and recall comparing with existing algorithms.

Acknowledgment. This material is based upon work supported by the Key Technologies Research and Development Program of China Foundation under Grants No. 2012BAH38F05 and also supported by the Research Program Foundation of Minjiang University under Grants No. MYK16002. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Key Technologies Research and Development Program of China Foundation and the Research Program Foundation of Minjiang University

REFERENCES

[1] A. Latham, K. Crockett, D. Mclean, An adaptation algorithm for an intelligent natural language tutoring system, *Computers & Education*, vol. 71, no. 1, pp. 97-110, 2014.

[2] M. Li, H. Liu, description of natural language based on human behaviour human body model semantic 3D,*Computer Applications and Software*, vol.02, pp. 177-181, 2014.

[3] X. j. Ye, Y. Yin, Natural Language Semantic Representation Based on Cognitive Linguistics, *Computer Science*, vol. 41, no. S1, pp. 98-102, 2014.

[4] G. Sidorov, F. Velasquez, E. Stamatatos, et al, Syntactic N-grams as machine learning features for natural language processing, *Expert Systems with Applications*, vol. 41, no. 3, pp. 853-860, 2014.

[5] Imler T D, Morea J, Kahi C, et al, Multi-center colonoscopy quality measurement utilizing natural language processing, *American Journal of Gastroenterology*, 2015, 110, no. 4), pp. 543-52.

[6] J. F. Cao, J. J. Chen, C. Yang, Image emotional semantic retrieval based on natural language understanding *Journal of Hunan University of Science & Technology , no. Natural Science Edition)*, no. 02, pp. 81-85, 2014.

[7] J. Xu,Maximum Fuzzy Marginal Projection via Patch Alignment Framework, *Computer Science*, vol. 43, no. 01, pp. 282-285, 2016.

[8] X. G. Jiang, P. Zhang, M. Sheng, Y. Li. Hu, Fuzzy Boundary Object Recognition Based on Cascading and Integrated Attribute Morphological Filtering, *Computer Engineering*, 42, no. 03, pp. 220-225 2016.

[9] J. A. K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, *Neural processing letters*, vol. 9, no. 3, pp. 293-300, 1999.

[10] J. S. Zhou, W. G. Qu, J. H. Xu, Y. Long, Y. B. Zhu, Implementing NLIs to GISs Using Semantic Parsing, *Journal of Chinese Information Processing*, vol. 28, no. 06, pp. 62-69, 2014.

[11] S. R. Eddy , Hidden markov models, *Current opinion in structural biology*, vol. 6, no. 3, pp. 361-365, 1996.

[12] Y. Feng, H. Q. Zhang, W. N. Hao, Novel conditional random field model extended by tensor and its application in natural language processing tasks, *Application Research of Computers*, vol. 33, no. 05, pp. 1428-1432, 2016.

[13] G. E. Hinton, S. Osindero, and Y.-W. Teh, A fast learning algorithm for deep belief nets, *Neural computation*, vol. 18, pp. 1527-1554, 2006.

[14] Y. Bengio, Learning deep architectures for AI, *Foundations and trends® in Machine Learning*,vol. 2, pp. 1-127, 2009.

[15] G. E. Hinton and R. S. Zemel, Autoencoders, minimum description length, and Helmholtz free energy *Advances in neural information processing systems*, pp. 3-3, 1994.

[16] Y. LeCun and Y. Bengio, Convolutional networks for images, speech, and time series, *The handbook of brain theory and neural networks*, vol. 3361, 1995.

[17] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in Acoustics, *Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013*, pp. 6645-6649.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, *arXiv preprint*,arXiv, pp. 1409-4842, 2014.

[19] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint*, arXiv, pp. 1409-1556, 2014.

[20] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, Deepface, Closing the gap to human-level performance in face verification, *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 1701-1708, 2014.

[21] J. Weston, S. Chopra, and A. Bordes, Memory networks, *arXiv preprint*, arXiv, pp. 1410.3916, 2014.

[22] G. E. Dahl, D. Yu, L. Deng, and A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, pp. 30-42, 2012.

[23] A. Fischer and C. Igel, An introduction to restricted Boltzmann machines, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, ed: Springer*, pp. 14-36, 2012.

[24] G. E. Hinton, Training products of experts by minimizing contrastive divergence, *Neural computation*, vol. 14, pp. 1771-1800, 2002.

[25] Miao Y, Metze F, Improving low-resource CD-DNN-HMM using dropout and multilingual, *DNN training*, 2013.