Information Fusion 70 (2021) 88–99

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Phonetic-enriched text representation for Chinese sentiment analysis with reinforcement learning

Haiyun Peng [a], Yukun Ma [a], Soujanya Poria [b], Yang Li [c], Erik Cambria [d,*]

[a] Alibaba Group, Singapore
[b] Singapore University of Technology and Design, Singapore
[c] Northwestern Polytechnical University, China
[d] Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Sentiment analysis
Multilingual sentiment analysis
Chinese phonetics
Deep phonemic orthography

ABSTRACT

The Chinese pronunciation system offers two characteristics that distinguish it from other languages: deep phonemic orthography and intonation variations. In this paper, we hypothesize that these two important properties can play a major role in Chinese sentiment analysis. In particular, we propose two effective features to encode phonetic information and, hence, fuse it with textual information. With this hypothesis, we propose Disambiguate Intonation for Sentiment Analysis (DISA), a network that we develop based on the principles of reinforcement learning. DISA disambiguates intonations for each Chinese character (pinyin) and, hence, learns precise phonetic representations. We also fuse phonetic features with textual and visual features to further improve performance. Experimental results on five different Chinese sentiment analysis datasets show that the inclusion of phonetic features significantly and consistently improves the performance of textual and visual representations and surpasses the state-of-the-art Chinese character-level representations.

1. Introduction

In recent years, sentiment analysis has become increasingly popular for processing social media data on online communities, blogs, wikis, microblogging platforms, and other online collaborative media [1]. It is a field of natural language processing (NLP) and affective computing [2] that comprises many subtasks, such as anaphora resolution [3] and subjectivity detection [4], and aims to classify text – but sometimes also audio, images, and video [5] – into positive, negative and neutral [6]. Sentiment analysis techniques can be broadly categorized into unsupervised [7], semi-supervised [8] and supervised [9] approaches.

While most approaches in the literature address the problem in a language-independent approach [10–12], Chinese sentiment analysis in fact requires tackling language-dependent challenges due to its unique nature, including word segmentation [13,14] and compositional analysis [15–19]. There are two main characteristics distinguishing Chinese from other languages. Firstly, it is a *pictogram* language [20], which means that symbols (called Hanzi) intrinsically carry meanings. Multiple symbols might form a new single symbol via geometric composition. The hieroglyphic nature of Chinese writing system differs from many Indo-European languages such as English or German. It has therefore inspired many works to explore the sub-word components (such as Chinese character and Chinese radicals) via a textual approach [15–19,21]. The other research line models the compositionality using the visual presence of the characters [22,23] by the means of extracting visual features from bitmaps of Chinese characters to further improve the Chinese textual word embeddings.

The second characteristic of Chinese is that it is a language of deep phonemic orthography according to the orthographic depth hypothesis [24,25]. In other words, it is hard to support the recognition of words involving language phonology. Each symbol of modern Chinese language can be phonetically transcribed into a romanized form, termed pinyin, consisting of an initial (optional), a final, and the tone. More specifically, as a tonal language, one single syllable in modern Chinese can be pronounced with five different tones, i.e., four main tones and one neutral tone (shown later in Table 5). We argue that this particular form of Chinese language provides semantic cues complementary to its textual form as illustrated in Table 1. It is worth noting that the tones alone do not have a direct link to sentiment polarities. Instead, tones' co-occurrence with specific syllable and context determines sentiment polarity. Despite its important role in Chinese language, to the best of our knowledge, it has not yet been explored by existing work for NLP tasks of Chinese language.

We surmise that the second factor of Chinese language can play a vital role in Chinese natural language processing especially sentiment analysis. Particularly, to consider the deep phonemic orthography and

* Corresponding author.
E-mail address: cambria@ntu.edu.sg (E. Cambria).

https://doi.org/10.1016/j.inffus.2021.01.005
Received 26 February 2019; Received in revised form 6 January 2021; Accepted 10 January 2021
Available online 14 January 2021

Table 1
Examples of intonations that alter meaning and sentiment.

Text	Pronunciation	Meaning	Sentiment polarity
空	kōng	Empty	Neutral
	kòng	Free	Neutral
假	jiǎ	Fake	Neutral/Negative
	jià	Holiday	Neutral
好吃	hǎochi	Delicious	Positive
	hàochi	Gluttony	Negative

intonation variety of the Chinese language, we propose two steps to learn Chinese phonetic information.

Firstly, we come up with two types of phonetic features. The first type extracted audio features from real audio clips. The second type learned pinyin token embeddings from a converted pinyin corpus. For each type of the features, we provide one version with intonation and one version without intonation.

Upon building the feature lookup table between each Chinese pinyin and its feature/embedding, we reach our second step, which is to design a Disambiguate Intonation for Sentiment Analysis (DISA) network that works on pinyin sequence and automatically decides the correct intonation for each pinyin. This step is crucial in disambiguating meanings and even sentiment of Chinese characters. Specifically, inspired by [26], we employ a reinforcement network as the main structure for our DISA network. The actor network is a typical neural policy network, whose action is to choose one out five intonations for each pinyin. The critic network is a long short-term memory (LSTM) sequence model, which learns the pinyin sentence sequence representation. The policy network is updated by a delayed reward when the sequence representation is built, while the critic network is updated by a sentiment class cross-entropy loss.

Motivated by the recent success of multimodal learning, we also incorporate textual and visual features with phonetic features. To the best of our knowledge, we are the first to consider the deep phonemic orthographic characteristic and intonation variation in a multimodal framework for the task of Chinese sentiment analysis. Experimental results show that the proposed multimodal framework outperforms the state-of-the-art Chinese sentiment analysis method by a statistically significant margin. In summary, the three main contributions of this paper are:

- We fuse different modalities associated with Chinese characters to emulate the way humans read and understand Chinese language.
- We introduce a reinforcement learning based framework, which jointly disambiguates intonations of Chinese characters and resolves sentiment ambiguity.
- We demonstrate the effectiveness of our framework on several benchmark datasets.

The remainder of this paper is organized as follows: we first present a brief review of embedding features, sentiment analysis and Chinese phonetics; we then introduce our model and provides technical details; next, we describe the experimental results and presents analytical discussions; finally, we conclude the paper and suggest future work.

2. Related work

We start with a brief review of textual embedding methods, followed by analyzing existing Chinese representations that include both textual embeddings and visual embeddings. Next, we briefly review sentiment analysis and Chinese phonetics.

2.1. Chinese Textual embedding

One-hot representation is the initial numeric word representation method in NLP. However, it usually leads to a problem of high dimensionality and sparsity. To solve this problem, distributed representation (or word embedding) [27] is proposed. Word embedding is a representation which maps words into low dimensional vectors of real numbers by using neural networks. The key idea is based on distributional hypothesis so as to model how to represent context words and the relation between context words and target word.

Since the introduction of C&W model [28] in 2008, direct learning of word embedding has become popular. In 2013, Mikolov et al. [29] introduced both Continuous Bag-of-words model (CBOW) and Skip-gram model. The former placed context words in the input layer and target word in the output layer whereas the latter swapped the input and output in CBOW. In 2014, Pennington et al. [29] created the 'GloVe' embeddings. Unlike the previous which learned the embeddings from minimizing the prediction loss, GloVe learned the embeddings with dimension reduction techniques on co-occurrence counts matrix.

2.2. Chinese Representation

Chinese text differs from English text for two key aspects: it does not have word segmentations and it has a characteristic of compositionality due to its pictogram nature. Based on the former aspect, word segmentation tools are always employed before text representation, such as ICTCLAS [30], THULAC [31], Jieba[1] and so forth. Based on the latter aspect, several works had focused on the use of sub-word components (such as characters and radicals) to improve word embeddings. Chen et al. [21] proposed decomposition of Chinese words into characters and presented a character-enhanced word embedding model (CWE). Sun et al. [15] and Li et al. [16] decomposed Chinese characters to radicals and developed a radical-enhanced Chinese character embedding. In [17], pure radical based embeddings were trained for short-text categorization, Chinese word segmentation and web search ranking. Yin et al. [18] extend the pure radical embedding by introducing multigranular Chinese word embeddings. Peng et al. [19] and Chao et al. [32] explored utilizing radical information for the sentiment analysis task. Recently, Zhang et al. [33] proposed an interactive multitask learning method for Chinese text sentiment classification.

Multimodal representation in the past few years has become a growing area of research. Liu et al. [23] and Su et al. [22] explored integrating visual features to textual word embeddings. The extracted visual features proved to be effective in modeling the compositionality of Chinese characters.

2.3. Sentiment analysis and Chinese phonetics

Sentiment analysis has raised growing interest both within the scientific community, leading to many exciting open challenges, as well as in the business world, due to the remarkable benefits to be had from business intelligence [34], political forecasting [35], recommender systems [36], social network analysis [37], dialog systems [38], and more. Various directions have been actively explored in the past few year, from document level [39–41], to sentence level [42–44] and to aspect level [45–47]. Most methods took a high perspective to develop effective models for a broad spectrum of languages [48]. Only a limited number of works spend efforts in studying language-specific characteristics [16,22,49]. Among them, there is almost no literature trying to take advantage of phonetic information for Chinese representation. We, however, believe the Chinese phonetic information could be of great value to the representation and sentiment analysis of Chinese language, due to but not limited to the following evidence.

[1] http://github.com/fxsjy/jieba

Shu and Anderson conducted a study on Chinese phonetic awareness in [50]. The study involved 113 participants of Chinese 2nd, 4th, and 6th graders enrolled in a working-class Beijing, China elementary school. Their task was to represent the pronunciation of 60 semantic–phonetic compound characters. Results showed that children as young as 2nd graders are better able to represent the pronunciation of regular characters than irregular characters or characters with bound phonetics.

The strong influence of familiarity on pronunciation underlines an unavoidable fact about the Chinese writing system: the system does not offer pronunciation cues that are as reliable or consistent as those of many other writing systems, such as English [51]. Moreover, Hsiao and Shillcock argued that semantic–phonetic compound (or phonetic compound) comprised about 81% of the 7000 frequent Chinese characters [52]. These compounds would affect semantics greatly if we can find an approach to effectively represent their phonetic information.

To the best of our knowledge, no previous work has integrated pronunciation information to Chinese representation. Due to its deep phonemic orthography, we surmise that the Chinese pronunciation information could elevate the representations to a higher level. Thus, we propose to learn phonetic features and present a DISA network to automatically convert the Chinese character to its pinyin with correct intonation.

3. Model

In this section, we first present how features from textual and visual modalities were extracted. Next, we delve deep into the details of different type of phonetic features. Then, we introduce a DISA network which parses Chinese characters to their pronunciations with tones. Lastly, we demonstrate how we fuse features from three modalities for sentiment analysis.

3.1. Textual embedding

As in most recent literature, textual word embedding vectors were treated as the fundamental representation of texts [27,29,53]. Firstly introduced by Bengio et al. [27], low-dimensional word embedding vectors learned a distributed representation for words. Compared with traditional n-gram word representations, they largely reduced the data sparsity problem and provided more friendly access towards neural networks. In 2013, Mikolov et al. [29] introduced the toolkit 'Word2Vec' which populated the application of word embedding vectors due to its fast learning time. In the toolkit, two predictive-based word vectors, CBOW and Skip-gram, were proposed. They either predicted the target word from context or vice versa. Pennington et al. [53] developed 'GloVe' in 2014 which employed a count-based mechanism to embed word vectors. Following the convention, we used 'GloVe' character embeddings [53] of 128-dimension to represent text.

It is worth noting that we set the fundamental token of Chinese text as the character instead of the word for two reasons. Firstly, the character is designed to align against the audio feature. Audio features can only be extracted at character level, as Chinese pronunciation is on each character. In Chinese language, the fundamental phonetic unit which is semantically self-contained is at character level. In English, however, the fundamental phonetic unit is at word level (except some prefix/suffix syllables). Secondly, character-level processing can avoid the errors induced by Chinese word segmentation. Although we used character GloVe embedding as our textual embedding, experimental comparisons were conducted with both CBOW [29] and Skip-gram embeddings.

Table 2
Configuration of convAE for visual feature extraction.

Layer#	Layer configuration
1	Convolution 1: kernel 5, stride 1
2	Convolution 2: kernel 4, stride 2
3	Convolution 3: kernel 5, stride 2
4	Convolution 4: kernel 4, stride 2
5	Convolution 5: kernel 5, stride 1
Feature	Extracted visual feature: (1,1,512)
6	Dense ReLu: (1,1,1024)
7	Dense ReLu: (1,1,2500)
8	Dense ReLu: (1,1,3600)
9	Reshape: (60,60,1)

Fig. 1. Original input bitmaps (upper row) and reconstructed output bitmaps (lower row).

3.2. Training visual features

Unlike the Latin language, the Chinese written language originated from pictograms. Afterwards, simple symbols were combined to form complex symbols in order to express abstract meanings. For example, a geometric combination of three '木 (wood)' creates a new character '森 (forest)'. This phenomenon gives rise to a compositional characteristic of Chinese text. Instead of a direct modeling of text compositionality using sub-word [18,21] or sub-character [15,16,19] elements, we opt for a visual model. In particular, we constructed a convolutional auto-encoder (convAE) to extract visual features (visual feature is not the main focus of this work). Details of the convAE are listed in Table 2.

Following the convention in [54] and [22], we set the input of the model to a 60 by 60 bitmap for each of the Chinese characters and the output of the model to a dense vector with a dimension of 512. The model was trained using Adagrad optimizer on the reconstruction error between original bitmap and reconstructed bitmap. The loss is given as:

$$\sum_{j=1}^{L}(|x_t - x_r| + (x_t - x_r)^2) \tag{1}$$

where L is the number of samples. x_t is the original input bitmap and x_r is the reconstructed output bitmap. An example of the original and reconstructed bitmaps is shown in Fig. 1. After training the visual features, we obtained a lookup table where each Chinese character corresponds to a 512-dimensional feature vector.

3.3. Learning phonetic features

Written Chinese and spoken Chinese have several fundamental differences. To the best of our knowledge, all the previous literature on Chinese NLP ignored the significance of the audio channel. As cognitive science suggests, human communication depends not only on visual recognition but also audio activation. This drove us to explore the mutual influence between the audio channel (pronunciation) and textual representation.

Popular Latin and Germanic languages such as Spanish, Portuguese, English etc. share two remarkable characteristics. Firstly, they have shallow phonemic orthography.[2] In other words, the pronunciation of a word is largely dependent on the text composition in such languages. One can almost infer the pronunciation of a word given its

[2] http://en.wikipedia.org/wiki/Phonemic_orthography

Table 3
Illustration of 4 types of phonetic features: a(x) stands for the extracted audio feature for pinyin 'x'; v(x) represents learned embedding vector for 'x'; number 0 to 4 represents 5 diacritics.

Text		假设明天放假。
English		Suppose tomorrow is holiday.
Pinyin		Jiǎ Shè Míng Tiān Fàng Jià
Extracted from audio	Ex0	a(Jia) a(She) a(Ming) a(Tian) a(Fang) a(Jia)
	Ex04	a(Jiǎ) a(Shè) a(Míng) a(Tiān) a(Fàng) a(Jià)
Learned from corpus	PO	v(Jia) v(She) v(Ming) v(Tian) v(Fang) v(Jia)
	PW	v(Jia3) v(She4) v(Ming2) v(Tian1) v(Fang4) v(Jia4)

textual spelling. From this perspective, textual information can be interchangeable with phonetic information.

For instance, if the pronunciations of English word 'subject' and 'marineland' were known, it is not hard to speculate the pronunciation of word 'submarine', because one can combine the pronunciation of 'sub' from 'subject' and 'marine' from 'marineland'. This implies that phonetic information of these languages may not have additional information entropy than textual information. Secondly, intonation information is limited and implicit in these languages. Generally speaking, emphasis, ascending intonation and descending intonation are the major variations in these languages. Although they exerted great influence in sentiment polarity during communication, there is no apparent clue to infer such information only from text [55].

However, Chinese language differs from the above-mentioned languages in several key aspects. Firstly, it is a language of deep phonemic orthography. One can hardly infer the pronunciation of Chinese word/character from its textual writing. For example, the pronunciations of characters '日' and '月' are 'rì' and 'yuè', respectively. A combination of the two characters makes another character '明' which pronounced 'míng'. This characteristic motivates us to find how the pronunciation of Chinese can affect natural language understanding. Secondly, intonation information of Chinese is rich and explicit. In addition to emphasis, each Chinese character has one tone (out of five different tones), marked by diacritics explicitly. These intonations (tones) greatly affect the semantic and sentiment of Chinese characters and words. Examples were shown in Table 1.

To this end, we found it was not trivial to explore how Chinese pronunciation can influence natural language understanding, especially sentiment analysis. In particular, we designed two approaches to learn phonetic information, namely feature extraction from audio signal and embedding vector learning from textual corpus. For either of the above two approaches, we have two variations, namely with (Ex04, PW) or without (Ex0, PO) intonations. An illustration is shown in Table 3. Details of each type will be introduced in the following sections.

3.3.1. Extracted feature from audio clips (*Ex0, Ex04*)

The spoken system of modern Chinese is named 'Hanyu Pinyin', abbreviated to 'pinyin'.[3] It is the official romanization system for mandarin in mainland China [56]. The system includes four diacritics denoting four different tones plus one neutral tone. For each of the Chinese characters, it has one corresponding pinyin. This pinyin has five variations in tones (we treat the neutral tone as one special tone). The statistics of Chinese character and pinyin are listed in Table 4. It shows that the number of frequently used characters is bigger than the number of pinyin counterparts with or without tones. This suggests that certain Chinese characters share the same pinyin and further implies that the one-hot dimensionality will reduce if pinyin was used to represent text.

In order to extract phonetic features, for each tone of each pinyin, we collected an audio clip which recorded a female's pronunciation of that pinyin (with tone) from a language learning resource.[4] Each audio clip lasts around one second with a standard pronunciation of one pinyin with tone. The quality of these clips was validated by two native speakers. Next, we used openSMILE [57] to extract phonetic features on each of the obtained pinyin-tone audio clip. Audio features are extracted at 30 Hz frame-rate and a sliding window of 20 ms. They consist of a total number of 39 low-level descriptors (LLD) and their statistics, e.g., MFCC, root quadratic mean, etc.

Table 4
Statistics of Chinese characters and 'Hanyu Pinyin'.

	Pinyin		Textual character
	w/o tones	w/ tones	
Number of tokens	374	1870	3500

After obtaining features for each of the pinyin-tone clip, we obtained an $m \times 39$ dimensional matrix for each clip, where m depends on the length of clip and 39 is the number of features. To regulate the feature representation for each clip, we conducted singular value decomposition (SVD) on the matrices to reduce them to 39-dimensional vectors, where we extracted the vector with the singular values. In the end, high dimensional feature matrices of each pinyin clip were transformed to a dense feature vector of 39 dimensions. A lookup table between pinyin and audio feature vector is constructed accordingly.

In particular, we prepared two sets of extracted phonetic features. The first type comes with tone, which is the feature we obtained from the above processing. We denote it as **Ex04**, where 'Ex' stands for extracted features and '04' stands for having one tone from 0 to 4 (we represent neutral tone as 0 and the first to the fourth tone as 1 to 4 respectively). The second type removed the variations of tones, in which we take the arithmetic mean of five features from five tones of each pinyin. We denote it as **Ex0**, where '0' stands for no tone. In the second type of feature, pinyin scripts with different tones will have same phonetic features, even though they may mean different meanings.

3.3.2. Learned feature from pinyin corpus (*PO, PW*)

Instead of collecting audio clips for each pinyin and extracting audio features, we directly represent Chinese characters with pinyin tokens, as shown in Table 3. Specifically, we convert each Chinese character in a textual corpus to it pinyin. The original corpus which was represented by a sequence of Chinese characters was converted to a phonetic corpus which was represented by a sequence of pinyin tokens.

In the phonetic corpus, contextual semantics were still maintained as in textual corpus. This is achieved with the help of online parser,[5] which parse Chinese characters to their pinyin counterparts. It should be pointed out that 3.49% of the common 3500 Chinese characters (around 122 characters) [58] have multiple pinyin scripts,[6] namely 'duo yin zi'(heteronym). Although the parser claimed its support to heteronym, we took the most statistically-possible pinyin prediction of each heteronym.

We did not disambiguate various heteronyms particularly, as this is not the major assumption we try to argue in this paper. However, it could be a direction worth working on in the future. The DISA provides two modes in its conversion from Character to pinyin, one with tone and the other without tone.

[3] http://iso.org/standard/13682.html

[4] http://chinese.yabla.com — This resource has only four tones for each pinyin, which does not have the neutral tone pronunciation. To obtain the neutral tone feature, we compute the arithmetic mean of the features of the other four tones.

[5] http://github.com/mozillazg/python-pinyin

[6] http://https://zh.m.wikisource.org/zh-hans/%E6%99%AE%E9%80%9A%E8%AF%9D%E5%BC%82%E8%AF%BB%E8%AF%8D%E5%AE%A1%E9%9F%B3%E8%A1%A8

For the mode without tone, Chinese characters will be converted to pinyin without tones only. Examples are the tokens shown in the row of PO in Table 3, where **PO** stands for Pinyin w/o tones. Afterwards, we train 128-dimension pinyin token embedding vectors using conventional 'GloVe' character embeddings [53]. A lookup table between pinyin without intonation (**PO**) and embedding vector is constructed accordingly. Pinyin scripts that have the same pronunciation but different intonations will share the same glove embedding vector, such as *Jiǎ* and *Jià* in Table 3.

For the mode with tone, Chinese characters will be converted to pinyin plus a number suggesting the tone. Examples are the tokens shown in the row of PW in Table 3, where **PW** stands for Pinyin w/ tones. We use number 1 to 4 to represent four diacritics and number 0 to represent the neutral tone. Similarly, 128-dimension 'GloVe' pinyin embedding vectors were trained.

In summary, we have four types of phonetic features, namely **Ex04**, **PW**, **Ex0** and **PO**. PO distinguishes from PW in removing intonations. Two of them (**Ex04**, **PW**) distinguish from others by having intonations. It is expected to have one question that how would one know the correct intonation of pinyin scripts given their textual characters. Although the online parser can give its statistical guess, the performance and robustness cannot be evaluated and guaranteed. To address this problem, we design a parser network with a reinforcement learning model to learn the correct intonation of each pinyin. Details will be presented in the following section.

3.4. DISA

3.4.1. Overview

This DISA network takes a sentence of Chinese characters as input. It firstly converts each character to its corresponding pinyin (without tones) through a lookup operation. Then the pinyin sequence will be fed to an **actor–critic network**. For each pinyin (time step), a policy network will randomly sample one out of five actions, where each action denotes a tone. Then a feature/embedding of this specific pinyin with tone is retrieved from a feature lookup module.

During exploration stage, the action will be randomly sampled. During exploitation and prediction stages, the action will be the one with maximum probability given the policy. This feature/embedding sequence will then be fed to an LSTM network. Hidden states from the LSTM will pass back to policy network for guiding action selection. The final hidden state of the LSTM network will be fed to a softmax classifier to obtain a sentence sentiment class distribution. A log probability of ground-truth label will be treated as a delayed reward to tune the policy network. Finally, a cross entropy loss will be computed against the obtained sentiment class distribution to tune the critic network. A graphical description is shown in Fig. 2, followed by details below.

State: For the environment, we used an LSTM to simulate the value function (detailed later). The input to this LSTM is the sequence of feature/embedding retrieved from the lookup module (detailed later), namely $x_1, x_2, \ldots, x_t, \ldots, x_L$, where x_t is the feature for the t*th* pinyin in the sentence. The mathematical representations of the LSTM cell are as follows:

$$\begin{aligned} f_t &= \sigma(W_f[x_t, h_{t-1}] + b_f) \\ I_t &= \sigma(W_I[x_t, h_{t-1}] + b_I) \\ \widetilde{C}_t &= tanh(W_C[x_t, h_{t-1}] + b_C) \\ C_t &= f_t * C_{t-1} + I_t * \widetilde{C}_t \\ o_t &= \sigma(W_o[x_t, h_{t-1}] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{2}$$

where f_t, I_t and o_t are the forget gate, input gate and output gate, respectively. W_f, W_I, W_o, b_f, b_I and b_o are the weight matrix and bias scalar for each gate. C_t is the cell state and h_t is the hidden state output.

The state of the environment is defined as:

$$S_t = [x_t \oplus h_{t-1} \oplus C_{t-1}] \tag{3}$$

where $\oplus$ is a concatenation (same below). As shown in Formula (3), the state is determined by the current feature input, the last LSTM hidden output and the last LSTM cell memory.

Action: There are five actions in our environment, representing five different tones. An example is shown in Table 5. If different action was selected, then the corresponding intonation will be activated. Relevant phonetic features will then be selected, as introduced in Section 3.4.3. The action policy was implemented by a typical feedforward neural network. Specifically, for a policy $\pi(a_t \mid S_t)$ at time t,

$$\pi(a_t \mid S_t) = \tanh(W \cdot S_t + b) \tag{4}$$

where W and b are the weight matrix and bias scalar. a_t is the action at time t. During exploration of training, action will be randomly selected out of the above five. During exploitation of training and testing, the action with the maximum probability will be selected.

Reward: The reward is computed at the end of each sentence when the state/action trajectory comes to the terminal (delayed reward). After the feature/embedding lookup module, the feature sequence is fed to the LSTM critic network. A sentence sentiment class distribution is computed as:

$$distr = \sigma(W_{sfmx} \cdot h_L + b_{sfmx}) \tag{5}$$

where W_{sfmx} and b_{sfmx} are weight matrix and bias scalar from the softmax layer. h_L is the last hidden state output from the LSTM critic network. $distr^{1*X}$ is the probability distribution of sentiment classes for the sentence. X is the number of sentiment class. The reward (R) is defined as:

$$R = log(P(ground \mid sent)) \tag{6}$$

where $P(ground \mid sent)$ stands for the probability of the ground-truth label of the sentence given the distribution in Eq. (5).

Table 5
Actions in DISA network and meanings.

Action	0	1	2	3	4
Intonation	Neutral	ˉ	ˊ	ˇ	ˋ
Example	a	ā	á	ǎ	à

3.4.2. Actor: policy network

As shown in the 'Action' above, the policy network random guesses actions during the exploration stage in training. It will be updated when a sentence input is fully traversed. Given the reward obtained from Eq. (6), we used gradient descent method to optimize the policy network [59]. In other words, we want to maximize:

$$\begin{aligned} J(\theta) &= E_\pi[R(S_1, a_1, S_2, a_2, \ldots, S_L, a_L)] \\ &= \sum_1^L p(S_1) \prod_t \pi_\theta(a_t \mid S_t) p(S_{t+1} \mid S_t, a_t) R_L \\ &= \sum_1^L \prod_t \pi_\theta(a_t \mid S_t) R_L \end{aligned} \tag{7}$$

Using the likelihood ratio (or REINFORCE [60] trick) to estimate policy gradient, the gradient can be transformed to:

$$\nabla_\theta J(\theta) = \sum_{t=1}^L R_L \nabla_\theta log \pi_\theta(a_t \mid S_t) \tag{8}$$

3.4.3. Feature/embedding lookup

Recall that we have selected actions from actor network, where each action denotes a tone for that pinyin, the function of this feature/embedding lookup module is to retrieve the correct features of that specific pinyin with tone. Prior to the policy network, we have collected phonetic features from five different tones of each pinyin and order them from neutral tone feature to the fourth tone feature. The

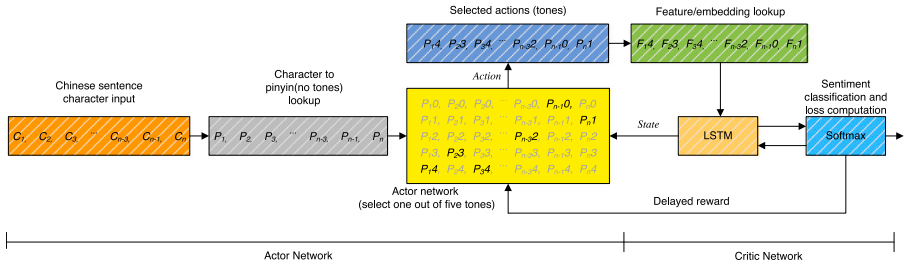


Fig. 2. DISA model structure for tone selection. C_m stands for the mth Chinese character in a sentence. P_m denotes the pinyin for mth character without the tones. $P_m n$ represents the pinyin for mth character with its nth tone. $F_m n$ is the feature/embedding vector for $P_m n$.

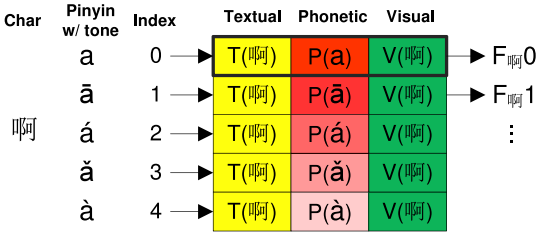


Fig. 3. An example of fused character feature/embedding lookup, where T, P, V represent features/embeddings from corresponding modality. In the case of single modality or bi-modality, relevant lookup table is constructed accordingly.

neutral tone to the fourth tone feature can be retrieved individually by index ID number 0 to 4.

When an action is selected from the actor network, for example, action 4 was selected for pinyin P_1, this lookup module will find the fourth phonetic feature (index ID 4) of this pinyin, namely $F_1 4$ and pass it to the LSTM critic network as the input x_t in Eq. (2).

3.4.4. Critic: sentence model and loss computation

Introduced in the **State** before, the critic network was essentially a sentence encoding model by an LSTM. We used gradient descent method to update the critic network with the cross-entropy loss defined as:

$$L = -\sum_{\forall sent} P(ground \mid sent) log(P(pred \mid sent)) \tag{9}$$

where $P(ground \mid sent)$ and $P(pred \mid sent)$ are the ground truth and predicted probability in the Eq. (5), respectively.

3.5. Fusion of modalities

In the context of the Chinese language, textual embeddings have been applied in various tasks and proved its effectiveness in encoding semantics or sentiment [15–18,21,61,62]. Recently, visual features pushed the performance of textual embedding further via a multimodal fusion [22,23,63]. This is achieved due to the effective modeling of compositionality of Chinese characters by the visual features. In this work, we hypothesize that the use of phonetic features along with textual and visual can improve the performance. Thus, we introduced the following fusion method that fits with our DISA network, as in Fig. 2.

- Each Chinese character is represented by a concatenation of three segments. Each segment represents one modality, see below:

$$char = [emb_T \oplus emb_P \oplus emb_V] \tag{10}$$

where $char$ is character representation. emb_T, emb_P, emb_V are embeddings from text, phoneme and vision, respectively.

Table 6
# of reviews in experimental datasets.

	Weibo	It168	Chn2000	Review-4	Review-5
Positive	1900	560	600	1975	2599
Negative	1900	458	739	879	1129
Sum	3800	1018	1339	2854	3728

There are other complex fusion methods available in the literature [64–66], however, we did not use them in our paper for three reasons — (1) Fusion through concatenation is often proven effective method [23,67,68], (2) it has the added benefit of simplicity, thus allowing for the emphasis (contributions) of the system to remain with the features themselves and (3) The designed fusion needs to fit in with our reinforcement model framework. Fusion methods as in [22] and [64] impose obstacles in the implementation with actor–critic model. Thus, we used the above introduced fusion method, an example of a fused feature/embedding lookup table is shown in Fig. 3.

4. Experiments and results

In this section, we start with introducing the experimental setup. Experiments were conducted in six steps. Firstly, we compare unimodal features. Secondly, we experiment on the possible fusion of modalities. Thirdly, we compare cross domain validation performance between our method with baselines. Next, we conduct ablation tests to validate the contribution of phonetic features. More precisely, we also visualize different phonetic features/embeddings to understand how they improve the performance.

4.1. Experimental setup

4.1.1. Datasets and features/embeddings

Datasets: We evaluate our method on five datasets: Weibo, It168, Chn2000, Review-4 and Review-5. The first three datasets consist of reviews extracted from micro-blog and review websites. The last two datasets contain reviews from [69], where Review-4 has reviews from computer and camera domains, and Review-5 contains reviews from car and cellphone domains. The experimental datasets are shown in Table 6.

Features/embeddings: For textual embeddings, we refer to the pretrained character embedding lookup table trained with Glove in Section 3.1. For phonetic experiments, we employ a pre-built tool called online codes[7] on the datasets to convert text to pinyin without intonations (As we discussed in Section 3.3.2, this conversion achieves as high as 97% accuracy.). Ex0 and Ex04 features were extracted from audio files and stored as in Section 3.3.1. PO and PW embeddings were also pretrained on the same textual corpus for training textual embeddings. The corpus contains news of 8 million Chinese words, which is equal to 38 million Chinese characters. For visual features,

[7] http://github.com/mozillazg/python-pinyin

we refer to the lookup table to convert characters to visual features as in Section 3.2.

For experiments of multimodality, features from each individual modality were concatenated into a lookup table. Examples are shown in Fig. 3.

4.1.2. Setup and baselines

Setup: We use TensorFlow and Keras to implement our model. All models use an Adam Optimizer with a learning rate of 0.001 and an L2-norm regularizer of 0.01. Dropout rate is 0.5. Each mini-batch contains 50 samples. We randomly split each dataset to training, testing and development sets per the ratio 6:2:2. We report the result of testing set whose corresponding development set performs the best after 30 epochs. The above parameters were set with the use of a grid search on the development data. We also experiment different splits of datasets, whose results did not vary obviously.

The training procedure of our DISA network is as follows. Firstly, we skip the policy network and directly train the LSTM critic network with the training objective as Eq. (9). Secondly, we fix the parameters of the LSTM critic network and train the policy network with the training objective as Eq. (8). Lastly, we co-train all the modules together until convergence. For the cases when no phonetic feature/embedding is involved, for example pure textual or visual features, only the LSTM is trained and tested. Glove was chosen as the textual embedding in our model due to its performance in Table 7.

DISA variants: We introduce below the variants of our DISA network. They differ in text representation features.

1. **DISA (P)**: DISA network that used phonetic feature only, which is the concatenation of Ex04 and PW.
2. **DISA (T+P)**: DISA network that uses the concatenation of textual embedding (glove) and phonetic feature (Ex04+PW).
3. **DISA (P+V)**: DISA network that uses the concatenation of phonetic feature (Ex04+PW) and visual feature.
4. **DISA (T+P+V)**: DISA network that uses the concatenation of textual embedding (glove), phonetic feature (Ex04+PW) and visual feature.

Baselines: Our proposed method is based on input features/embeddings of Chinese characters. A handful number of the works on Chinese text analysis aimed at improving Chinese word embeddings, such as CWE [21], MGE [18]. Those who utilized visual features [22,23] also aimed at word level. However, these cannot be considered as comparable fair baselines to our model, our model is set up with Chinese character embedding being its core foundation. There are two major reasons for studying at character level. Firstly, pinyin pronunciation system is designed for character level. The pinyin system does not have corresponding pronunciations to Chinese words. Secondly, character level will bypass Chinese word segmentation operation which may induce errors. Conversely, using character-level pronunciation to model word-level pronunciation will incur sequence modeling issues. For instance, a Chinese word '你好' is comprised of two characters, '你' and '好'. For textual embedding, the word can be treated as one single unit by training a word embedding vector. For phonetic embedding, however, we cannot treat the word as one single unit from the perspective of pronunciation. The correct pronunciation of the word is a time sequence of character pronunciation of firstly '你' and then '好'. If we work at word level, we have to come up with a representation of the pronunciation of this word, such as an average of character phonetic features etc. To make a fair comparison, we compare with character-level methods below (all trained on the same corpus and character embedding vector dimension was set to 128.):

1. **Glove**: An unsupervised embedding learning algorithm based on co-occurrence (count). [53].
2. **CBOW**: Continuous Bag-of-words model which places context words in the input layer and target word in the output layer [29].
3. **Skip-gram**: The opposite of CBOW model, which predicts the contexts given the target word [29].
4. **Visual**: Based on [22] and [54], a convolutional auto-encoder (convAE) is built to extract compositionality of Chinese characters through the visual channel.
5. **charCBOW**: Component-enhanced character embedding built on top of CBOW method by [16]. It delved into the radical components of Chinese characters and enriched the character representation with radical component.
6. **charSkipGram**: The Skip-gram variant of charCBOW.
7. **Hsentic**: Radical-based hierarchical embeddings, which encode both radical- and character-level semantics (specifically tuned for sentiment analysis [19]).

Table 7
Classification accuracy of unimodality in LSTM.

		Weibo	It168	Chn2000	Review-4	Review-5
GloVe		**75.39**	81.82	**84.54**	**87.46**	**86.94**
CBOW		72.39	78.75	81.18	85.11	84.71
Skip-gram		75.05	80.13	78.04	86.23	86.21
Visual		61.78	65.40	67.21	78.98	79.59
charCBOW		71.54	80.83	82.82	86.90	85.19
charSkipGram		71.86	82.10	81.63	85.21	84.84
Hsentic		73.65	80.23	79.09	84.76	73.31
Phonetic features	DISA(Ex04)	67.28	84.69	78.18	81.88	83.38
	DISA(PW)	67.80	83.73	77.45	85.37	84.18
	DISA(P)	68.19	**85.17**	79.27	84.67	85.24

4.2. Experiments on unimodality

For textual embeddings, we have compared with state-of-the-art embedding methods including GloVe, skip-gram, CBOW, charCBOW, charSkipGram and Hsentic. As shown in Table 7, textual embeddings (GloVe) achieve the best performance among all three modalities in four datasets. This is due to the fact that they successfully encoded the semantics and dependency between characters. We also find that charCBOW and charSkipGram methods perform quite close to the original CBOW and Skip-gram methods. They perform slightly but not constantly better than their baselines. We conjecture this could be caused by the relatively small size of our training corpus compared to the original Chinese Wikipedia Dump training corpus. With the corpus size increased, all embedding methods are expected to have improved performance. It is without doubts, though, that the corpus we used still presents a fair platform for all methods to compare.

We also notice that visual feature achieves the worst performance among three modalities, which is within our expectations. As demonstrated in [22], pure visual features are not representative enough to obtain a comparable performance with the textual embedding. Last but not least, our methods with phonetic features perform better than the visual feature. Although visual features capture compositional information of Chinese characters, they fail to distinguish different meanings of characters that have same writing but different tones. These tones could largely alter the sentiment of Chinese words and further affect sentiment of sentence.

As for phonetic representation, three types of features were tested, namely Ex04, PW and P (namely Ex04+PW). The last one is the concatenation of the previous two. Our first observation is that phonetic features alone can hardly compete with textual embeddings. Although they beat textual embeddings in It168 dataset, they consistently fell behind textual embeddings. This is still within our expectations, as suggested by Tseng in [70], 'Phonology and phonetics alone are insufficient in predicting the actual output of sentences'.

If we further refer to Table 4, we can find that on average 2 to 3 characters share one same pinyin with tone. That means a pure phonetic representation may wipe out the 1 out of 2 or 3 (33%–50%)

Table 8
Classification accuracy of multimodality. (T and V represent textual and visual, respectively. + means the fusion operation. P is the concatenated phonetic feature of the one extracted from audio (Ex04) and Pinyin w/ intonation (PW).)

	Weibo	It168	Chn2000	Review-4	Review-5
GloVe	75.39	81.82	84.54	87.46	86.94
Visual	61.78	65.40	67.21	78.98	79.59
charCBOW	71.54	80.83	82.82	86.90	85.19
charSkipGram	71.86	82.10	81.63	85.21	84.84
Hsentic	73.65	80.23	79.09	84.76	73.31
DISA(P)	68.19	85.17	79.27	84.67	85.24
DISA(T+P)	**75.75**	**86.12**	**85.45**	**90.42**	**90.03**
DISA(T+V)	73.79	85.65	83.27	89.37	88.70
DISA(P+V)	76.01	82.30	81.09	86.76	87.23
DISA(T+P+V)	74.32	77.99	78.18	87.63	89.49

semantics from the text. This inevitably will reduce the possibility to correctly classify the sentiment.

As we can see each modality has its own capacity to encode semantics, it is expected to take advantage of the complimentary information from multiple modalities for the sentiment analysis task. The results are shown in the next section.

4.3. Experiments on fusion of modalities

In this set of experiments, we evaluate the fusion of every possible combination of modalities. After extensive experimental trials, we summarize that the concatenation of Ex04 and PW embeddings (denoted as P) performed best among all phonetic feature combinations. Thus we use it as phonetic feature in the fusion of modalities. The results shown in Table 8 suggest that the best performance is achieved by fusing textual and phonetic features.

We notice that phonetic features when fused with textual or visual features, improve the performance of both textual and visual unimodal classifiers consistently. This validates our hypothesis that phonetic features are an important factor in representing semantics, which leads to an improvement in Chinese sentiment analysis performance. A p-value of 0.007 in the paired t-test between with and without phonetic features suggested that the best performing improvement of integrating phonetic feature is statistically significant. The integration of multiple modalities could take advantages of information from different modalities. However, we notice that, in most of the cases, tri-modal models underperform bi-modal models. One disadvantage brought by using more modalities is the increase of number of parameters. We conjecture that a larger set of learnable parameters leads to poor generalizability when the training sets in our experiments only consist of instances of less than 4000.

Furthermore, the information redundancy becomes more severe when combining features across different modalities. In other words, there might be the marginal effect of using additional modality. We will illustrate this point with examples. As aforementioned, Chinese character is made of symbols (or called radicals). Some symbols function as morphemes, while some symbols function as phonemes. For instance, the character '疯' consists of two symbols, '疒' and '风'. The pronunciation of '疯' (feng1) is dominated by the symbol '风' (feng1), which is the same for phonetic features. Meanwhile, '风' contributes the most to the visual image of '疯', the visual feature of '疯' can also somehow encodes the information brought by '风'.

After we compare T with T+P and T+V, the performance increase induced by P is 1.40% higher than by V on average. It is apparent to conclude that phonetic feature is better at encoding semantics than visual features. The fusion of phonetic and textual embeddings achieve the best performance in all of the cases. It indicates that the information encoded in the phonetic feature complements that of textual embedding.

Table 9
Cross-domain evaluation. Datasets on the first column are the training sets. Datasets on the first row are the testing sets. The second column represents various baselines and our proposed method.

		Weibo	It168	Chn2000	Review-4	Review-5
Weibo	Hsentic	-	66.47	61.84	64.93	63.71
	charCBOW		67.55	64.08	62.09	67.78
	charSkipGram		65.29	59.60	53.22	49.49
	DISA(T+P)		**73.68**	**66.55**	**69.16**	**71.01**
It168	Hsentic	59.15	-	59.30	69.76	67.62
	charCBOW	57.54		65.05	72.25	68.13
	charSkipGram	54.54		64.68	68.19	64.38
	DISA(T+P)	**63.75**		**68.36**	**77.00**	**74.07**
Chn2000	Hsentic	56.36	60.67	-	52.03	44.77
	charCBOW	56.23	70.40		61.77	63.36
	charSkipGram	51.99	68.53		62.47	62.77
	DISA(T+P)	**60.50**	**68.90**		**68.64**	**69.02**
Review-4	Hsentic	58.15	73.55	59.22	-	80.55
	charCBOW	54.91	72.96	58.40		80.77
	charSkipGram	54.65	71.88	65.27		80.31
	DISA(T+P)	**58.15**	**77.51**	**65.45**		**88.70**
Review-5	Hsentic	58.44	74.73	69.08	83.15	-
	charCBOW	56.73	72.47	57.06	85.77	
	charSkipGram	56.44	75.32	66.77	83.67	
	DISA(T+P)	**62.06**	**85.65**	**69.09**	**88.85**	

4.4. Cross-domain evaluation

In this section, we examine how our model performs across different domains and datasets in order to validate the generalizability of our proposed method. Particularly for our model, we firstly pretrain the LSTM critic network on the training set. Then we fix the parameters of critic network and train the policy network on the same training set. Next, we co-train the LSTM critic network and policy network for 30 epochs. For other baselines, an LSTM network is trained using the same training set. By the end of each epoch, the development set of this training dataset and the other four datasets are tested. The epoch results are recorded. In the end, the testing result of the epoch which has the best development result are reported. The final results of the state-of-the-art methods are shown in Table 9.

Results show that all methods reduce their performance compared to single dataset experiments due to the internal diversity of different dataset. Even though, our method still perform better than other baselines by an average of 6.50% in accuracy. In addition to absolute performance, we also compute the average performance loss for each method across different datasets between single dataset case and cross-dataset case. It shows that our method has the least performance drop, which is 14.25%. The performance drop for Hsentic, charCBOW and charSkipGram methods are 16.09%, 15.69%, 17.16% respectively. We think it might be ascribed to that the proportion of shared phonetic tokens among datasets is larger than the portion of shared textual characters (1870 phonetic tokens Vs. 3500 textual characters). Thus, phonetic features will have better transferability than textual features. Fig. 4 illustrates the proportion of common phonetic tokens as well as common textual tokens between each pair of datasets. The result in the figure agrees with our initial analysis. We also noted that cross-domain results fall similar to single-domain results when the training and testing tests tend to be similar domains.

4.5. Ablation tests

We conduct ablation tests in two steps: validating phonetic features and integration of phonetic features. The first step validates the contribution of phonetic features. The second step examines which specific combination of phonetic features works the best.

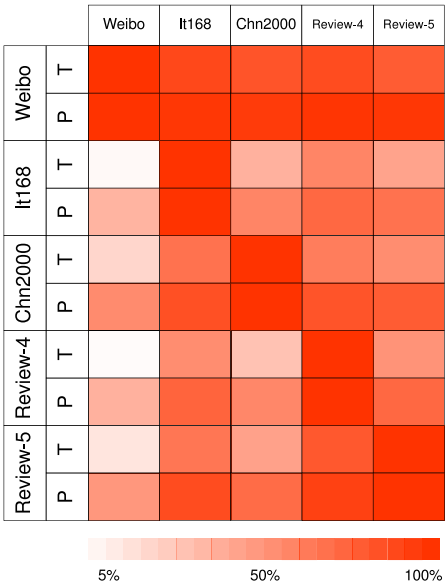


Fig. 4. The proportion of tokens in testing sets that also appear in training sets. Rows are training sets(T denotes the textual token and P denotes the phonetic token) Columns are testing sets.

Table 10
Performance comparison between learned and random generated phonetic feature.

		Weibo	It168	Chn2000	Review-4	Review-5
Random phonetic feature (rand)		53.83	56.85	55.71	69.20	69.77
Learned phonetic feature	Ex0	66.49	84.21	77.82	81.36	83.24
	Ex04	67.28	**84.69**	**78.18**	81.88	83.38
	PO	64.28	82.30	77.09	83.97	82.71
	PW	**67.80**	83.73	77.45	**85.37**	**84.18**

4.5.1. Validating phonetic feature

So far, we have examined the effectiveness of our model as a whole by comparing it with different baselines. In this section, we break down the proposed methods into a reinforcement learning framework and a set of features. First of all, we would like to validate if the performance gain mainly results from the reinforcement learning framework. To this end, we replace the phonetic features with random features. In particular, we generate random real-valued vectors as random phonetic feature for each character. Each dimension of the random phonetic feature vector is a float number between −1 to 1 sampled from a Gaussian distribution. Then, we use this random feature vector to represent each Chinese character and yielded the results in Table 10.

In the comparison between the learned phonetic feature and random phonetic feature, we can observe that the learned feature outperforms random feature with at least 13% in all datasets. This result indicates that the improvement of performance is due to the contribution of learned phonetic feature but not the training of classifiers. Phonetic feature itself is the cause and similar performance will not be achieved just by introducing random features.

We plot the results in Fig. 5 on the left to amplify the difference. Moreover, we find that, whether extracted from audio clips or learned from pinyin corpus, phonetic features that contain intonation (Ex04 and PW) perform better than those without intonation (EX0 and PO) in all our experiments.

This proves our initial argument that intonation plays an important role in representing Chinese sentiment. Nevertheless, we also discover that the performances of various learned phonetic features are not persistent. PW prevails in three datasets while Ex04 wins in the other two datasets. As the best two phonetic features are either extracted from audio clips or learned from pinyin corpus, it is expected to take the advantage of both sides. Thus we propose the ablation test of phonetic feature in different combination.

Table 11
Performance comparison between different combinations of phonetic features.

	Weibo	It168	Chn2000	Review-4	Review-5
Ex0	66.49	84.21	77.82	81.36	83.24
Ex04	67.28	84.69	78.18	81.88	83.38
PO	64.28	82.30	77.09	83.97	82.71
PW	67.80	83.73	77.45	**85.37**	84.18
Ex0+PO	65.45	81.82	77.09	83.98	83.38
Ex0+PW	67.80	82.30	78.91	84.84	84.71
Ex04+PO	67.14	80.38	77.45	83.80	84.84
Ex04+PW	**68.19**	**85.17**	**79.27**	84.67	**85.24**

4.5.2. Integration of phonetic features

We combine both extracted phonetic features and learned phonetic features to form four variations. The results are shown in Table 11 and plotted in Fig. 5 on the right.

As expected, the combination of Ex04 and PW prevails in four datasets and performs close to the best in the remaining dataset. Specifically, when we compare Ex04+PW with Ex04, there is an average improvement of 1.43% across datasets. We believe the improvement was due to the semantic information provided by PW feature, as it was trained on the pinyin corpus. Contextual relation was designed to be encoded in embeddings. By merging embedding features to extracted features, the combination feature would also encode certain semantics, which we would show in the following section. Correspondingly, if we compare Ex04+PW with PW, the performance improvement was 0.80% on average.

This would be explainable due to Ex04 features extracted information that can only be conveyed in pronunciation. As we introduced in the start, the deep phonemic orthography has enabled Chinese pronunciation to encode meanings that were not represented in the text. The English text, in contrast, originally was designed to mimic pronunciation [51]. Due to the heterogeneity between textual and phonetic representation of the Chinese language, it is reasonable to unveil the magic behind Chinese phonetics. In summary, we have shown that both the intonation variation and deep phonemic orthography contributed to Chinese sentiment analysis task.

4.6. Visualization

In this section, we visualize four kinds of phonetic-related embeddings. The are Ex04, PW, Ex04+PW (P) and T+P.

As shown in Fig. 6(a), pinyin that have similar pronunciations (vowels) are close to each other in the embedding space. This observation matches our experimental purpose that the Ex04 feature will encode phonetic information (such as similarity) among different pronunciations. Secondly, as can be seen in Fig. 6(b), we visualize the embeddings of PW. Since it was learned on the phonetic corpus, certain semantics are expected to be encoded. In reality, we do find semantic closeness in the visualization. The squares are some examples we spotted. For instance, ‘Niu2’ and ‘Nai3’ are together due to ‘Niu2 Nai3(milk)’. ‘Dian4’ and ‘Nao3’ are together due to ‘Dian4 Nao3 (computer)’. ‘Jian3’ and ‘Cha2’ are together due to ‘Jian3 Cha2 (inspection)’. Next, we visualize the combined embedding, Ex04+PW, which is also the main phonetic feature we use in our model in Fig. 6(c). Unsurprisingly, we observe that this feature combines the characteristics both from Ex04 and PW because of this embedding clusters not only phonetic similarity but also semantic similarity. Finally, we visualize the fused embedding of T+P in Fig. 6(d). In addition to the characteristics displayed in Ex04+PW (P), the fused T+P appends with Chinese textual characters. For example, ‘沐’ Mu4 and ‘浴’ Yu4 stayed together because of semantics(bath). ‘桓’ Huan2 and ‘寰’ Huan2 stayed together because of phonetics. It can be

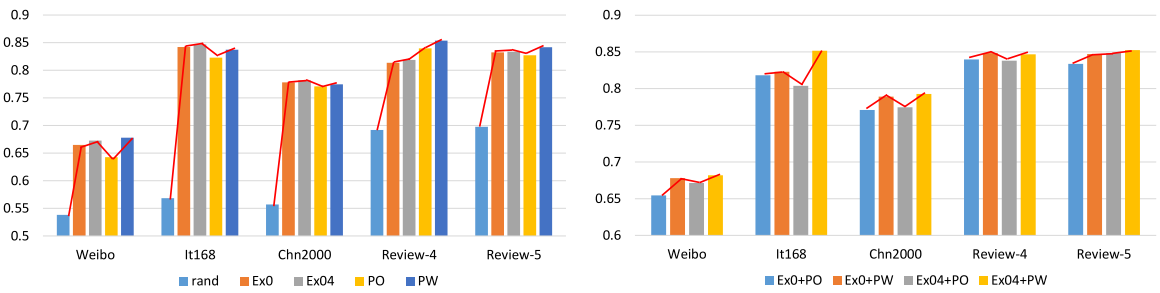
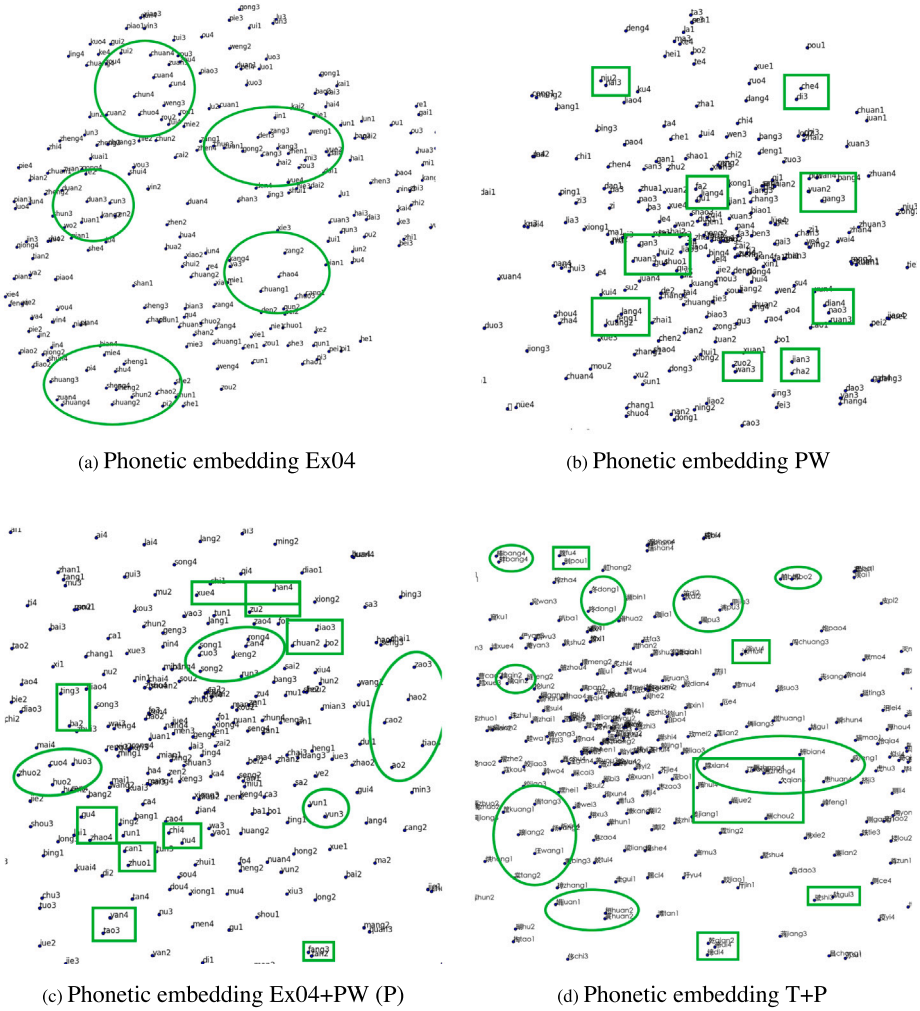


Fig. 5. Performance comparison between phonetic ablation test groups. Rand denotes random generated embeddings. Ex0/Ex04 represent Ex embeddings without/with tones. The same is for PO/PW. + denotes a concatenation operation.



(a) Phonetic embedding Ex04

(b) Phonetic embedding PW

(c) Phonetic embedding Ex04+PW (P)

(d) Phonetic embedding T+P

Fig. 6. Selected t-SNE visualization of four kinds of phonetic-related embeddings. Circles cluster phonetic similarity. Squares cluster semantic similarity.

concluded that the fused embeddings capture both certain phonetic information from phonetic features and semantic information from textual embeddings. These two sources of information when combine combined offers a finer and more accurate representation of Chinese characters, which is beneficial to sentiment analysis. This shows us why phonetic-enrich text representation could level up the performance in sentiment analysis compared with pure textual representation.

5. Conclusion

Modern Chinese pronunciation system (pinyin) provides a new perspective in addition to the standard pictogram-based writing system. Due to its deep phonemic orthography and intonation variations, it is expected to bring new contributions to the statistical representation of Chinese language, especially in complex NLP tasks like sentiment analysis.

To the best of our knowledge, we are the first to present an approach to learn phonetic information out of pinyin (both from audio clips and pinyin token corpus) and design a network to disambiguate intonations. We integrate phonetic information with textual and visual features to create novel multimodal representations for Chinese words. Experiments on five datasets portray the positive contribution of phonetic information to Chinese sentiment analysis.

Even though our method only examines Chinese language, it suggests greater potential for languages that also carry the deep phonemic orthography characteristic, such as Arabic and Hebrew. In the future, we plan to extend this work by exploring better fusion methods to combine different modalities and by also integrating word-level phonetic information.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] E. Cambria, H. Wang, B. White, Guest editorial: Big social data analysis, Knowl.-Based Syst. 69 (2014) 1–2.

[2] S. Poria, E. Cambria, R. Bajpai, A. Hussain, A review of affective computing: From unimodal analysis to multimodal fusion, Inf. Fusion 37 (2017) 98–125.

[3] R. Sukthanker, S. Poria, E. Cambria, R. Thirunavukarasu, Anaphora and coreference resolution: A review, Inf. Fusion 59 (2020) 139–162.

[4] I. Chaturvedi, E. Cambria, R. Welsch, F. Herrera, Distinguishing between facts and opinions for sentiment analysis: Survey and challenges, Inf. Fusion 44 (2018) 65–77.

[5] I. Chaturvedi, R. Satapathy, S. Cavallari, E. Cambria, Fuzzy commonsense reasoning for multimodal sentiment analysis, Pattern Recognit. Lett. 125 (264–270) (2019).

[6] A. Valdivia, V. Luzón, E. Cambria, F. Herrera, Consensus vote models for detecting and filtering neutrality in sentiment analysis, Inf. Fusion 44 (2018) 126–135.

[7] E. Cambria, Y. Li, F. Xing, S. Poria, K. Kwok, SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis, in: CIKM, 2020, pp. 105–114.

[8] A. Hussain, E. Cambria, Semi-supervised learning for big social data analysis, Neurocomputing 275 (2018) 1662–1673.

[9] M.S. Akhtar, A. Ekbal, E. Cambria, How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble, IEEE Comput. Intell. Mag. 15 (1) (2020) 64–75.

[10] W. Li, L. Zhu, Y. Shi, K. Guo, E. Cambria, User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models, Appl. Soft Comput. 94 (106435) (2020).

[11] I. Chaturvedi, S. Cavallari, E. Cambria, R. Welsch, Genetic programming for domain adaptation in product reviews, in: IEEE Congress on Evolutionary Computation, 2020.

[12] A. Valdivia, E. Martinez-Camara, I. Chaturvedi, M. Luzon, E. Cambria, Y.-S. Ong, F. Herrera, What do people think about this monument? Understanding negative reviews via deep learning, clustering and descriptive rules, J. Ambient Intell. Humanized Comput. 11 (1) (2020) 39–52.

[13] C. Huang, H. Zhao, Chinese word segmentation: A decade review, J. Chin. Inf. Process. 21 (3) (2007) 8–20.

[14] H. Peng, E. Cambria, A. Hussain, A review of sentiment analysis research in Chinese language, Cogn. Comput. 9 (4) (2017) 423–435.

[15] Y. Sun, L. Lin, N. Yang, Z. Ji, X. Wang, Radical-enhanced Chinese character embedding, in: LNCS, Vol. 8835, 2014, pp. 279–286.

[16] Y. Li, W. Li, F. Sun, S. Li, Component-enhanced Chinese character embeddings, 2015, arXiv preprint arXiv:1508.06669.

[17] X. Shi, J. Zhai, X. Yang, Z. Xie, C. Liu, Radical embedding: Delving deeper to Chinese radicals, Short Pap. 2 (2015) 594.

[18] R. Yin, Q. Wang, P. Li, R. Li, B. Wang, Multi-granularity Chinese word embedding, in: EMNLP, 2016, pp. 981–986.

[19] H. Peng, E. Cambria, X. Zou, Radical-based hierarchical embeddings for Chinese sentiment analysis at sentence level, in: FLAIRS, 2017, pp. 347–352.

[20] C. Hansen, Chinese ideographs and western ideas, J. Asian Stud. 52 (2) (1993) 373–399.

[21] X. Chen, L. Xu, Z. Liu, M. Sun, H. Luan, Joint learning of character and word embeddings, in: IJCAI, 2015, pp. 1236–1242.

[22] T.-r. Su, H.-y. Lee, Learning Chinese word representations from glyphs of characters, in: EMNLP, 2017, pp. 264–273.

[23] F. Liu, H. Lu, C. Lo, G. Neubig, Learning character-level compositionality with visual features, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2017, pp. 2059–2068.

[24] R. Frost, L. Katz, S. Bentin, Strategies for visual word recognition and orthographical depth: A multilingual comparison, J. Exp. Psychol. Hum. Percept. Perform. 13 (1) (1987) 104.

[25] L. Katz, R. Frost, The reading process is different for different orthographies: The orthographic depth hypothesis, Adv. Psychol. Amsterdam 94 (1992) 67.

[26] T. Zhang, M. Huang, L. Zhao, Learning structured representation for text classification via reinforcement learning, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 6053–6060.

[27] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (Feb) (2003) 1137–1155.

[28] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 160–167.

[29] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.

[30] H.-P. Zhang, H.-K. Yu, D.-Y. Xiong, Q. Liu, HHMM-based Chinese lexical analyzer ICTCLAS, in: Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Vol. 17, Association for Computational Linguistics, 2003, pp. 184–187.

[31] M. Sun, X. Chen, K. Zhang, Z. Guo, Z. Liu, Thulac: An Efficient Lexical Analyzer for Chinese, Tech. Rep., Technical Report, 2016.

[32] A.F. Chao, H.-L. Yang, Using Chinese radical parts for sentiment analysis and domain-dependent seed set extraction, Comput. Speech Lang. 47 (2018) 194–213.

[33] H. Zhang, S. Sun, Y. Hu, J. Liu, Y. Guo, Sentiment classification for Chinese text based on interactive multitask learning, IEEE Access 8 (2020) 129626–129635.

[34] Y. Li, S. Wang, Y. Ma, Q. Pan, E. Cambria, Popularity prediction on vacation rental websites, Neurocomputing 412 (2020) 372–380.

[35] A. Khatua, A. Khatua, E. Cambria, Predicting political sentiments of voters from Twitter in multi-party contexts, Appl. Soft Comput. 97 (106743) (2020).

[36] C. Angulo, Z. Falomir, D. Anguita, N. Agell, E. Cambria, Bridging cognitive models and recommender systems, Cogn. Comput. 12 (2) (2020) 426–427.

[37] D. Camacho, A. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, E. Cambria, The four dimensions of social network analysis: An overview of research methods, applications, and software tools, Inf. Fusion 63 (2020) 88–120.

[38] Y. Ma, K.L. Nguyen, F. Xing, E. Cambria, A survey on empathetic dialogue systems, Inf. Fusion 64 (2020) 50–70.

[39] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 1422–1432.

[40] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.

[41] I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, E. Cambria, Bayesian network based extreme learning machine for subjectivity detection, J. Franklin Inst. 355 (4) (2018) 1780–1797.

[42] Y. Kim, Convolutional neural networks for sentence classification, 2014, arXiv preprint arXiv:1408.5882.

[43] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Vol. 10, Association for Computational Linguistics, 2002, pp. 79–86.

[44] P. Chikersal, S. Poria, E. Cambria, A. Gelbukh, C.E. Siong, Modelling public sentiment in Twitter: Using linguistic patterns to enhance supervised learning, in: Computational Linguistics and Intelligent Text Processing, Springer, 2015, pp. 49–65.

[45] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, AAAI Press, 2017, pp. 4068–4074.

[46] H. Peng, Y. Ma, Y. Li, E. Cambria, Learning multi-grained aspect target sequence for Chinese sentiment analysis, Knowl.-Based Syst. 148 (2018) 167–176.

[47] Y. Ma, H. Peng, T. Khan, E. Cambria, A. Hussain, Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis, Cogn. Comput. 10 (4) (2018) 639–650.

[48] S.L. Lo, E. Cambria, R. Chiong, D. Cornforth, Multilingual sentiment analysis: From formal to informal and scarce resource languages, Artif. Intell. Rev. 48 (4) (2017) 499–527.

[49] S. Cao, W. Lu, J. Zhou, X. Li, cw2vec: Learning Chinese word embeddings with stroke n-gram information, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 5053–5061.

[50] H. Shu, R.C. Anderson, N. Wu, Phonetic awareness: Knowledge of orthography–phonology relationships in the character acquisition of Chinese children, J. Educ. Psychol. 92 (1) (2000) 56.

[51] K.H. Albrow, The English Writing System: Notes Towards a Description, Longman, 1972.

[52] J.H.-w. Hsiao, R. Shillcock, Analysis of a Chinese phonetic compound database: Implications for orthographic processing, J. Psycholinguist. Res. 35 (5) (2006) 405–426.

[53] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.

[54] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 52–59.

[55] Y. Xia, E. Cambria, A. Hussain, H. Zhao, Word polarity disambiguation using Bayesian model and opinion-level features, Cogn. Comput. 7 (3) (2015) 369–380.

[56] A. Benjamin, History and prospect of Chinese romanization, Chin. Librariansh. (1997).
[57] F. Eyben, M. Wöllmer, B. Schuller, Opensmile: the munich versatile and fast open-source audio feature extractor, in: Proceedings of the 18th ACM International Conference on Multimedia, ACM, 2010, pp. 1459–1462.
[58] X. Cao, Pu tong hua yi du ci shen yin, Zhongguoyuwen (1) (2002) 82–87.
[59] R.S. Sutton, D.A. McAllester, S.P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: Advances in Neural Information Processing Systems, 2000, pp. 1057–1063.
[60] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (3–4) (1992) 229–256.
[61] B. Ma, Q. Qi, J. Liao, H. Sun, J. Wang, Learning Chinese word embeddings from character structural information, Comput. Speech Lang. 60 (101031) (2020).
[62] Z. Wang, Z. Huang, J. Gao, Chinese text classification method based on BERT word embedding, in: ICMAI, 2020, pp. 66–71.
[63] H.-Y. Chen, S.-H. Yu, S.-d. Lin, Glyph2Vec: Learning Chinese out-of-vocabulary word embedding from glyphs, in: ACL, 2020, pp. 2865–2871.
[64] A. Zadeh, M. Chen, S. Poria, E. Cambria, L.-P. Morency, Tensor fusion network for multimodal sentiment analysis, in: EMNLP, 2017, pp. 1103–1114.
[65] N. Majumder, D. Hazarika, A. Gelbukh, E. Cambria, S. Poria, Multimodal sentiment analysis using hierarchical fusion with context modeling, Knowl.-Based Syst. 161 (2018) 124–133.
[66] E. Cambria, N. Howard, J. Hsu, A. Hussain, Sentic blending: Scalable multimodal fusion for continuous interpretation of semantics and sentics, in: IEEE SSCI, Singapore, 2013, pp. 108–117.
[67] C.G. Snoek, M. Worring, A.W. Smeulders, Early versus late fusion in semantic video analysis, in: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM, 2005, pp. 399–402.
[68] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
[69] W. Che, Y. Zhao, H. Guo, Z. Su, T. Liu, Sentence compression for aspect-based sentiment analysis, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (12) (2015) 2111–2124.
[70] C.-y. Tseng, An acoustic phonetic study on tones in Mandarin Chinese, Thesis (1983) 0438.