Learning Dimensional Sentiment of Traditional Chinese Words with Word Embedding and Support Vector Regression

河南工業大學
Henan University of Technology

Baoli Li
libaoli@gmail.com
Department of Computer Science, Henan University of Technology, Zhengzhou, China

What's the problem?

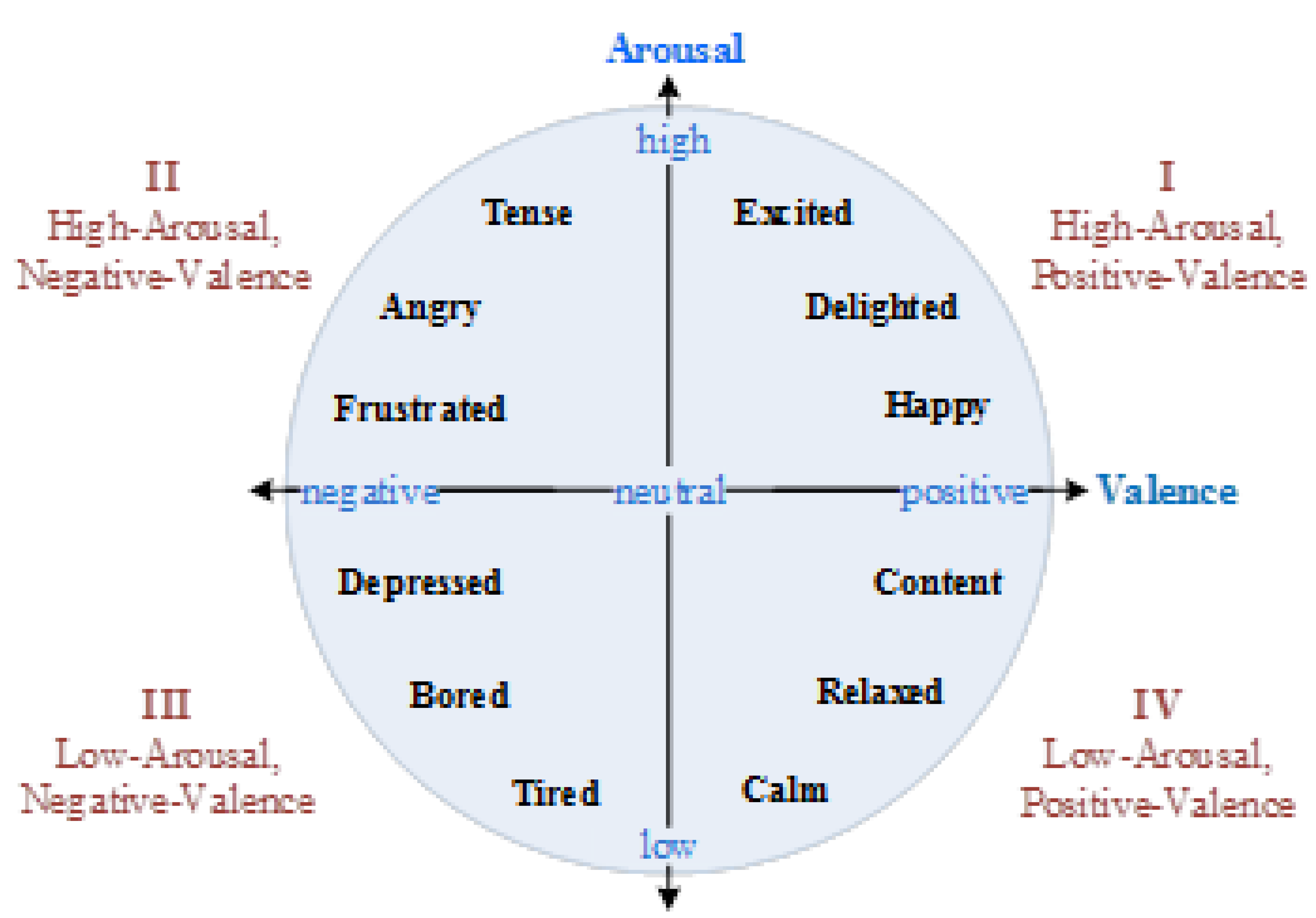
Dimensional sentiment analysis (DSA), which represents affective states as **continuous numerical** values on multiple dimensions, such as valence-arousal (VA) space, allows for more fine-grained analysis than the traditional categorical approach.

In dimensional sentiment analysis, **an affective lexicon enriched with dimensional sentiment values** is a key and necessary resource.

However, such a lexicon in a reasonable scale usually **does not exist**, and building such a lexicon **costs much**.

Motivation: targeting at the above problem, **how to expand a small seed lexicon automatically**.

Input: given a seed lexicon L, which contains V words and annotates each word with numerical valence-arousal ratings, a new word W_{new}, and other resources allowed (open).
Output: the VA ratings of the word W_{new}.



Two-dimensional valence-arousal space

The valence represents the degree of pleasant and unpleasant feelings, while the arousal describes the degree of excitement and calm.

Learning Word Dimensional Sentiment using Word Embedding and Support Vector Regression

1. *Build **a valence-arousal space (VAS)** with all the seed words: each dimension corresponds to a seed word*
2. *Position a (seed or new) word into a point in the VAS: the value at each dimension is determined by the similarity between the word and the seed word linked to that dimension. We use **word embedding** on a large traditional Chinese corpus to facilitate the similarity calculation.*
3. *Derive a training data set **TD** with the seed words: each sample from a seed word corresponds to a point in the VAS space, and its prediction value is the valence or arousal value of the seed word.*
4. *Train a **Support Vector Regression** model based on **TD**.*
5. *To predict the valence or arousal value of a new word, get the point of the new word in the VAS space, and make prediction with the trained SVR model.*

Experiments and Results

Example 1:
Input: 0002, 痛苦
Output: 0002, 2.4, 6.8
Example2:
Input: 0004, 放鬆
Output: 0004, 6.2, 2.0

Datasets

- **IALP2016 Shared Task**: The Chinese Valence-Arousal Words (CVAW) lexicon, which contains 1,653 affective words annotated with valence-arousal ratings (1-9), is given as training data. Average valence rating of the 1,653 words is **4.495**, and average arousal rating is **5.745**. The test data consists of 1,149 traditional Chinese words.

Settings

- Use **Word2vec** package for word embedding: CBOW model, negative sampling.
- Large corpus (**56,501,758 words**) of fictions and proses collected from the website http://www.txttw.com.
- For support vector regression, we use the popular **Libsvm** package.
- Mean absolute error (**MAE**) and Pearson correlation coefficient (**PCC**) for evaluation.

Results

Table 1. The official Results of the IALP-2016 shared task

Metric / System	Valence			Arousal		
	MAE	PCC	Rank by Mean Rank	MAE	PCC	Rank by Mean Rank
HAUT1	0.615	0.780	15	1.285	0.607	12
HAUT2	**0.605**	**0.778**	**12**	**1.278**	**0.609**	**7**
BASELINE	1.407	0.674	29	1.567	0.473	28
Best Result	0.577	0.865	N/A	0.953	0.671	N/A
Average	0.788	0.766	N/A	1.251	0.479	N/A
Worst Result	1.735	0.35	N/A	1.567	0.169	N/A
Baseline_A (5.0,5.0)	1.013	0	N/A	0.979	0	N/A
Baseline_B (4.5, 5.7)	1.028	0	N/A	1.423	0	N/A

Table 2 Performance with different size of embedding vectors

Metric / Size	Valence MAE	Valence PCC	Arousal MAE	Arousal PCC	MED
100	0.644	0.772	1.283	**0.616**	1.532
200	0.605	**0.778**	**1.278**	0.609	**1.512**
300	**0.604**	0.769	1.280	0.592	1.519
400	0.605	0.758	1.296	0.584	1.537
500	0.607	0.757	1.298	0.575	1.543

Table 3 Performance with different Min-count values

Metric / Min-Count	V-MAE	V-PCC	A-MAE	A-PCC	MED	#OOV_Tr	#OOV_Te
2	0.610	0.777	1.282	**0.613**	1.519	**16**	**5**
3	**0.605**	0.778	1.278	0.609	1.512	24	8
4	0.609	**0.782**	1.287	0.610	1.519	30	12
5	0.612	0.778	1.279	**0.619**	1.515	35	16
6	0.610	0.774	**1.255**	0.617	**1.497**	36	20
7	0.621	0.769	1.267	0.610	1.510	39	24
8	0.624	0.769	1.271	0.608	1.515	45	28
9	0.616	0.773	1.264	0.610	1.505	49	33
10	0.610	0.772	1.274	0.612	1.511	58	35