NUSTM

Lexicon Construction with Representation Learning Based on Hierarchical Sentiment Supervision

Leyi Wang, Rui Xia*

Text Mining Group

Nanjing University of Science & Technology

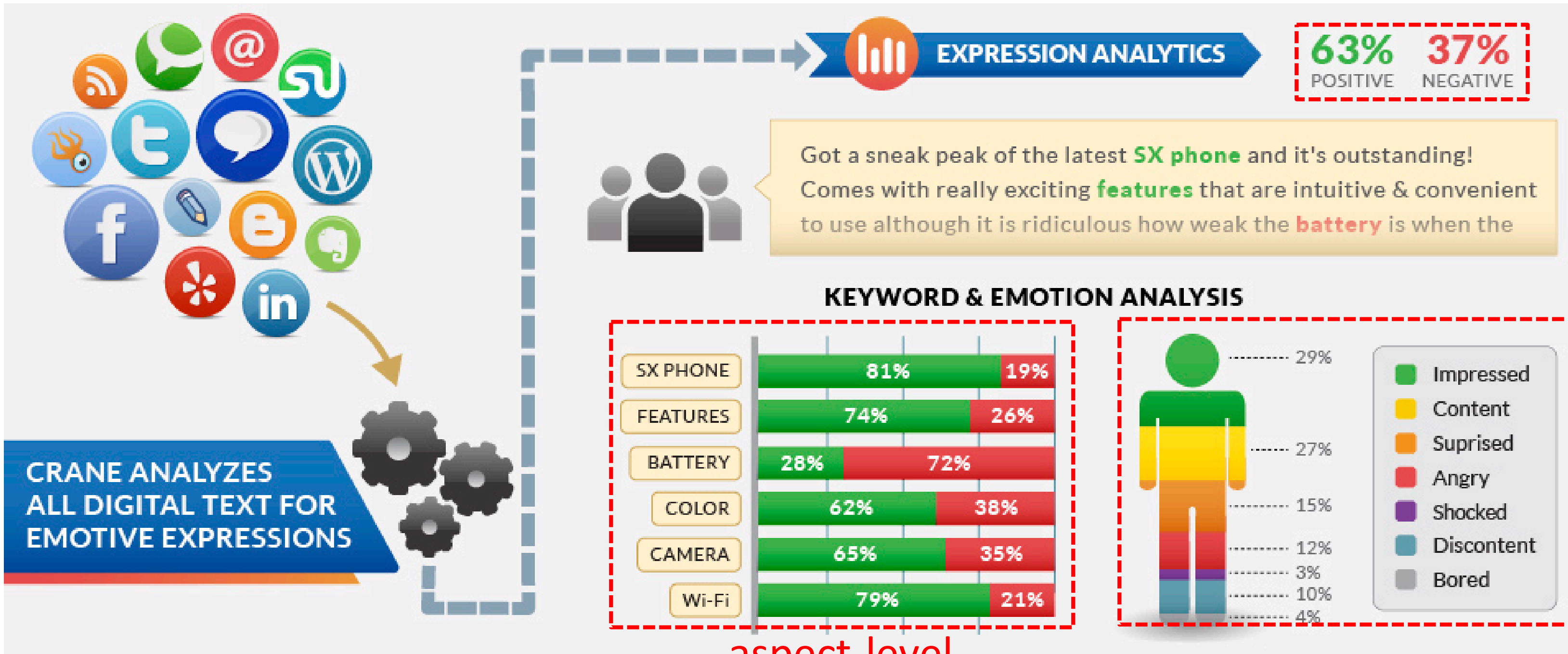
leyiwang.cn@gmail.com

Outline

- Introduction
- Our Approach
- Experiments
- Conclusions

Introduction

document-level



aspect-level

How to build sentiment lexicons

- Methods of Sentiment Lexicons Construction
  - Constructing sentiment lexicons manually
  - Constructing sentiment lexicons automatically
    - Dictionary-based methods
    - Corpus based methods
      - Conjunction Relation (by “and” or “but”)
      - *Co-occurrence (PMI)*
      - *Sentiment Representation Learning*

Challenges and Motivation

- Complex Linguistic Phenomena
  - Example

Four more fake people added me. Is this why people don't like Twitter? :(

- Sentiment Representation Learning

$$de = \frac{1}{|d|}\sum_{t \in d} e_t$$

So, all words are associated with the negative label.

- Challenges

Such linguistic phenomena occur frequently in review texts, and makes sentiment-aware word representation learning less effective.

Challenges and Motivation

- Our Motivation
  - Sentiment Representation Learning

Document-level: $de = \frac{1}{|d|}\sum_{t \in d} e_t$ → 🙂 **or** ☹️

Word-level: $e_t, \quad t \in d$ → 🙂 **or** ☹️

- Advantages

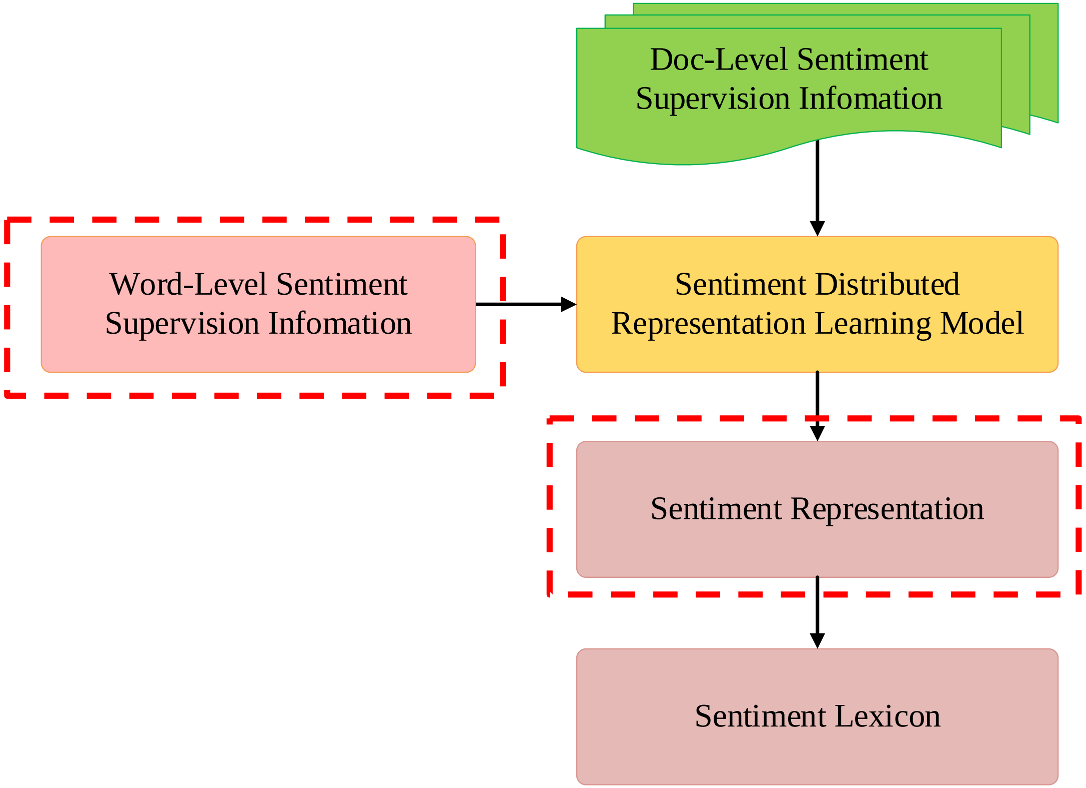
Avoiding the sentiment learning flaws caused by coarse-grained document-level supervision by incorporating fine grained word-level supervision.

Outline

- Introduction
- Our Approach
- Experiments
- Conclusions

Our Approach

- The Architecture of Our Method



Learning Word-Level Supervision

- PMI-based method

$$PMI(t,c) = \log \frac{p(t,c)}{p(t)\,p(c)} = \log \frac{p(t\,|\,c)}{p(t)}$$

- Sentiment Orientation

$$SO(t) = PMI(t,+) - PMI(t,-)$$

- Pseudo Sentiment Distribution

$$[\hat{p}(-\,|\,t), \hat{p}(+\,|\,t)] = \begin{cases} [0,1] & SO(t) > 0 \\ [1,0] & SO(t) < 0 \\ random\{[0,1],[1,0]\} & otherwise \end{cases}$$

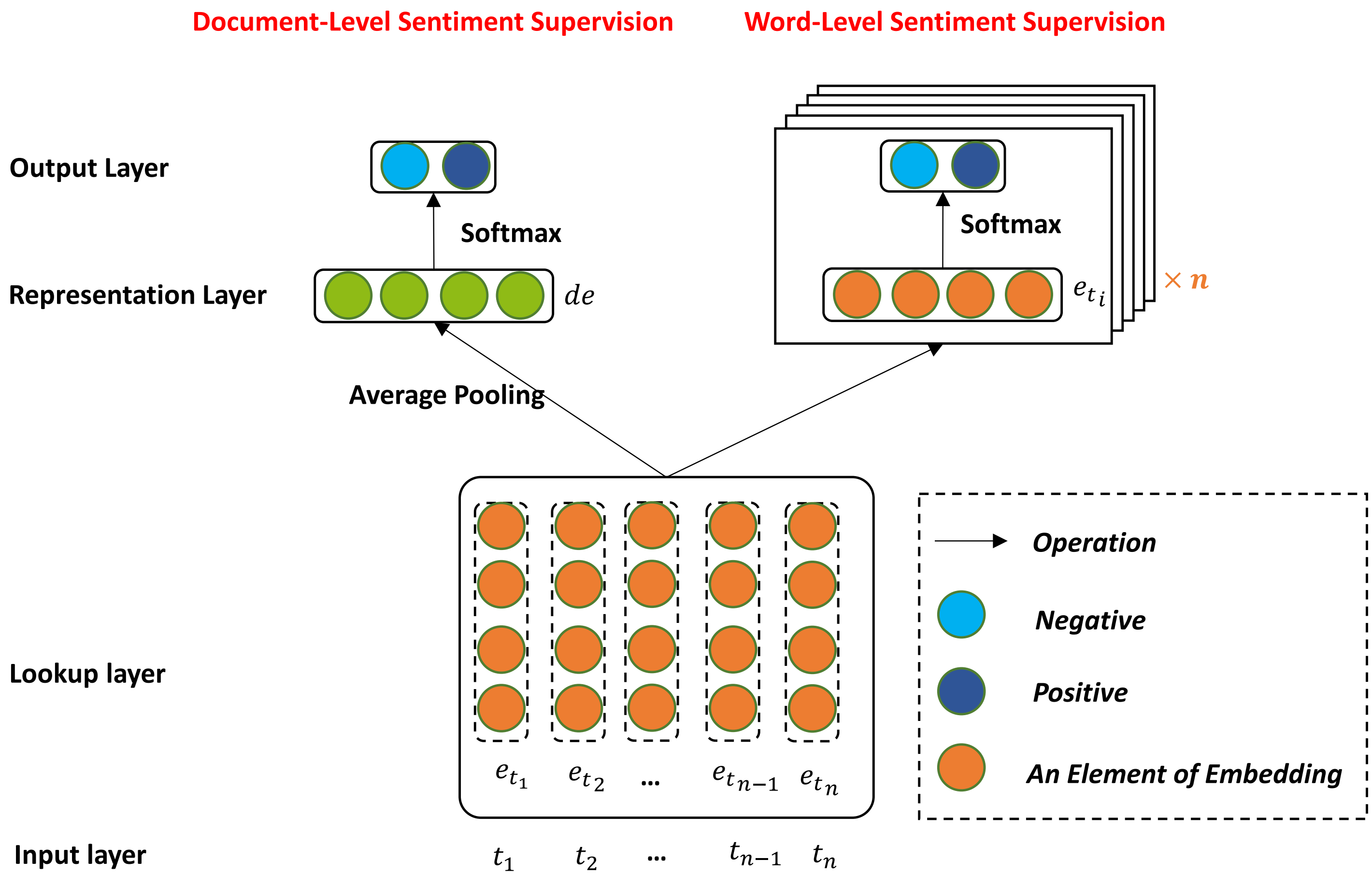
hard sentiment annotation

$$[\hat{p}(-\,|\,t), \hat{p}(+\,|\,t)] = [1-\sigma(SO(t)),\ \sigma(SO(t))]$$

soft sentiment annotation

Learning Sentiment Representation

- The Architecture of the Network



Learning Sentiment Representation

- Word-Level Sentiment Supervision
  - Hypothesis

$$[p(-|e_t), p(+|e_t)] = \text{softmax}(\theta_t \cdot e_t + b_t)$$

  - Cost Function

$$f_{word} = -\frac{1}{T}\sum_{k=1}^{N}\sum_{t \in d_k}\sum_{c \in \{-,+\}} \widehat{p}(c|t) \log p(c|e_t)$$

where N and T represent the number of documents and words in corpus, respectively.

- Document-Level Sentiment Supervision
  - Hypothesis

$$de = \frac{1}{|d|}\sum_{t \in d} e_t$$

$$[p(-|de), p(+|de)] = \text{softmax}(\theta_d \cdot de + b_d)$$

NUSTM

Learning Sentiment Representation

- Cost Function

$$f_{doc} = -\frac{1}{N}\sum_{k=1}^{N}\sum_{c\in\{-,+\}}\widehat{p}(c\,|\,d_k)\log p(c\,|\,de_k)$$

where N is the number of documents in corpus.

- Joint Cost Function

$$f = \alpha f_{word} + (1-\alpha) f_{doc}$$

Our goal is to minimize the joint cost function.

- Optimization

  \- Stochastic gradient descent

NUSTM

Building Sentiment Lexicon

- Seed Words

Negative	Positive
109	125

- Extend Seed Words on Urban Dictionary
  - Firstly, we utilize the embedding of 125 positive and 109 negative seed words manually labelled by Tang et al. (2014) as training data.
  - Secondly, a *variant-KNN* classifier is also applied to extending the seed words on a web dictionary called Urban Dictionary.

- Construct Sentiment Lexicon

$$[p(-|t), p(+|t)] = \mathrm{softmax}(\theta \cdot e_t + b)$$

$$score(t) = p(+|t) - p(-|t)$$

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014. Building large-scale twitter-specific sentiment lexicon : A representation learning approach. COLING 2014, pages 172–182.

Outline

- Introduction
- Our Approach
- Experiments
- Conclusions

Experiments

- Dataset and Settings

The statistics of representation learning datasets (Go et al., 2009)

Negative	Positive	Total
800,000	800,000	1600,000

The statistics of evaluation datasets

Dataset	#pos	#neg	Total
SemEval2013-train	3632	1449	5081
SemEval2013-dev	482	282	764
SemEval2013-test	1474	559	2033
SemEval2014-test	982	202	1184
SemEval2015-test	1038	365	1403
SemEval2016-test	7059	3231	10290

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1(2009):12.

Experiments

- Supervised Sentiment Classification Evaluation

  We report the performance of SVM by using these lexicon features. The *LIBSVM* toolkit is used with a linear kernel and the penalty parameter is set as the default value.

  Utilizing the lexicon features (for each polarity) (Mohammad et al, 2013)

  - total count of tokens in the tweet with score greater than 0;
  - the sum of the scores for all tokens in the tweet;
  - the maximal score;
  - the non-zero score of the last token in the tweet.

- Unsupervised Sentiment Classification Evaluation

  Sum up the scores of all sentiment words.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. Proceedings of the International Workshop on Semantic Evaluation.

Experiments

- (External) Comparison with Public Lexicons

Supervised Evaluation for External Evaluation (F_1 Score)

Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
Sentiment140	0.7317	0.7271	0.6917	0.6809	0.7079
HIT	0.7181	0.6947	0.6797	0.6928	0.6963
NN	0.7225	0.7115	**0.6970**	0.6887	0.7049
ETSL	0.7104	0.7090	0.6650	0.6862	0.6926
HSSWE	**0.7550**	**0.7424**	0.6921	**0.7097**	**0.7248**

HSSWE outperforms Sentiment140, HIT, NN and ETSL 1.7, 2.8, 1.9, and 3.2 percentages on the average of four datasets.

- **Sentiment140** was constructed by Mohammad et al. (2013) on tweet corpus based on PMI between each word and the emoticons.
- **HIT** was constructed by Tang et al. (2014) with a representation learning approach.
- **NN** was constructed by Vo and Zhang (2016) with a neural network method.
- **ETSL** refers to SemEval-2015 English Twitter Sentiment Lexicon (Rosenthal et al., 2015; Kiritchenko et al., 2014), which is done using Best-Worst Scaling.

NUSTM

Experiments

- (External) Comparison with Public Lexicons

Unsupervised Evaluation for External Evaluation (Accuracy)

Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
Sentiment140	0.7208	0.7416	0.6935	0.6928	0.7122
HIT	0.7566	0.7922	0.7128	0.7282	0.7474
NN	0.6903	0.7280	0.6507	0.6585	0.6819
ETSL	0.7675	0.8226	0.7505	**0.7365**	0.7693
HSSWE	**0.7734**	**0.8539**	**0.7669**	0.7206	**0.7787**

Across four datasets, the average accuracy of HSSWE is 6.6, 3.1, 9.6 and 0.94 higher than Sentiment140, HIT, NN and ETSL, respectively.

- **Sentiment140** was constructed by Mohammad et al. (2013) on tweet corpus based on PMI between each word and the emoticons.
- **HIT** was constructed by Tang et al. (2014) with a representation learning approach.
- **NN** was constructed by Vo and Zhang (2016) with a neural network method.
- **ETSL** refers to SemEval-2015 English Twitter Sentiment Lexicon (Rosenthal et al., 2015; Kiritchenko et al., 2014), which is done using Best-Worst Scaling.

Experiments

- (Internal) Comparison within the Model

Supervised Evaluation for Internal Evaluation (*F_1* Score)

Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
PMI-SO	0.7265	0.7333	0.7008	0.6858	0.7116
Doc-Sup	0.7326	0.7302	0.6814	0.6986	0.7107
HSSWE(soft)	**0.7550**	**0.7424**	0.6921	**0.7097**	**0.7248**
HSSWE(hard)	0.7503	0.7383	**0.7020**	0.7061	0.7242

Unsupervised Evaluation for Internal Evaluation (Accuracy)

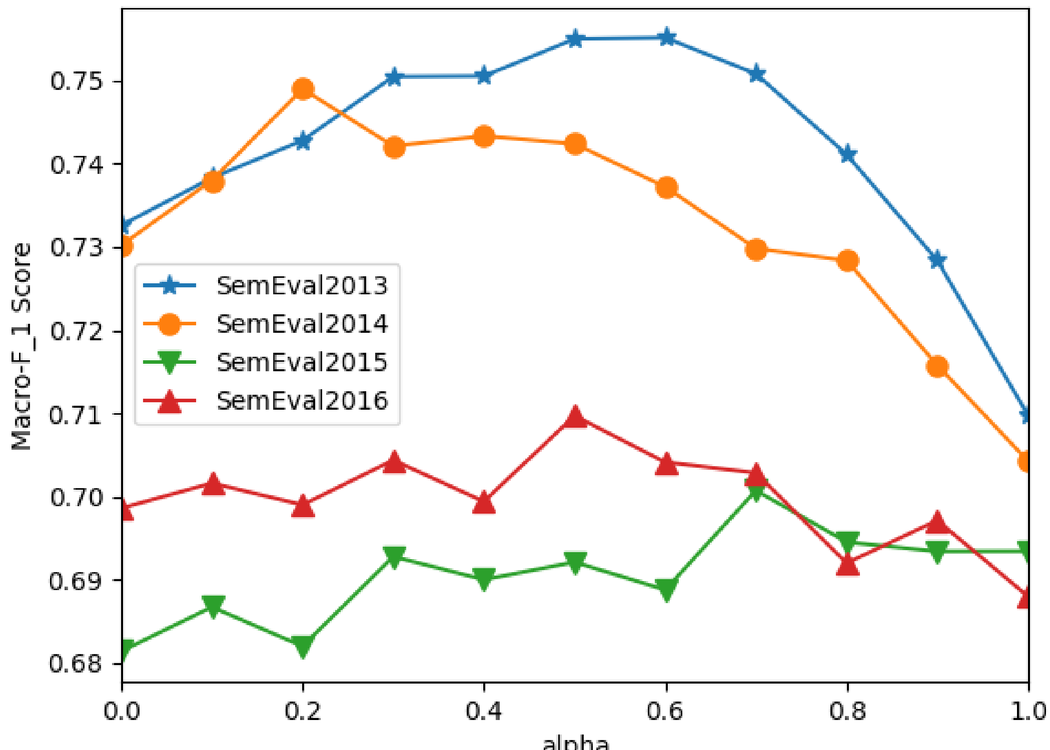
Lexicon	Semeval2013	Semeval2014	Semeval2015	Semeval2016	Average
Doc-Sup	0.7252	0.8294	0.7391	0.6859	0.7449
HSSWE(soft)	***0.7734***	***0.8539***	***0.7669***	***0.7206***	***0.7787***
HSSWE(hard)	0.7418	0.8395	0.7633	0.7011	0.7614

- **PMI-SO** denotes a PMI-SO based sentiment lexicon with soft sentiment annotation.
- **Doc-Sup** denotes the neural network system with only document-level sentiment supervision. It equals to HSSWE when $\alpha = 0$.
- **HSSWE(soft) and HSSWE(hard)** utilize the PMI-SO lexicon with soft and hard sentiment annotation at the word level, respectively.

NUSTM

Experiments

- The Effective of Parameter α



HSSWE performs better when is in the range of [0.45,0.55].

Outline

- Introduction
- Our Approach
- Experiments
- Conclusions

Conclusions

- We proposed to learn the sentiment-aware word representation under supervision at both document and word levels.

- HSSWE supports several kinds of word level sentiment annotations
  - Predefined sentiment lexicon
  - PMI-SO lexicon with hard sentiment annotation
  - PMI-SO lexicon with soft sentiment annotation.

- We verify the effectiveness of sentiment-aware word representation for sentiment lexicon construction.


THANK YOU