From CopeOpi Scores to CopeOpi Vectors: Word Vectors for Multi-class Text Classification

Pei-Shan Tsai

Advisor: Ying-ping Chen

Natural Computing Laboratory
Department of Computer Science
National Chiao Tung University

2017

Table of Contents

Table of Contents

Background

- Millions of digital texts are generated everyday. To derive useful information from these digital texts, text mining has become a popular area of both research and business
  - Text classification is one of the most important task
- Text classification, or text categorization
  - Assigning a document to a set of predefined classes, categories or labels
- In the past, text classification problems were solved by
  - Manually assignment
  - Knowledge engineering approaches (hand-crafted classification rules)
  - Both are expensive to scale due to the needs of skilled labors and expert knowledge
- Nowadays, works on classification focus on machine learning approaches

Definition of Text Classification

Definition (Text Classification)

In a text classification problem, we are given

- A document space $\mathbb{X}$
- A set of predefined classes $\mathbb{C}$

The task of text classification can be defined as an unknown assignment function

$$f : \mathbb{X} \times \mathbb{C} \rightarrow \{\texttt{True}, \texttt{False}\}$$

which assigns each pair $\langle d, c \rangle \in \mathbb{X} \times \mathbb{C}$ a Boolean value `True` if the document d is in the class c or `False` otherwise[1, 2]

Definition of Supervised Learning for Text Classification

Definition (Supervised Learning for Text Classification)

By using

- A machine learning algorithm Γ
- A labeled training set $\mathbb{D} = \{\langle d, c\rangle \,|\, \langle d, c\rangle \in \mathbb{X} \times \mathbb{C}\}$

We wish to learn a classifier, or classification function γ which approximates the unknown assignment function f as close as possible[3, 1, 2]

$$\Gamma(\mathbb{D}) = \gamma$$

$$\gamma : \mathbb{X} \times \mathbb{C} \to \{\texttt{True}, \texttt{False}\} \approx f$$

Applications

Typically, the document space $\mathbb{X}$ can be any kinds of texts and the classes $\mathbb{C}$ are defined for the user needs, thus text classification has a wide variety of applications in text mining

- Document organization and information retrieval
  - $\mathbb{X}$ = articles
  - $\mathbb{C}$ = topics
- Sentiment analysis and opinion mining
  - $\mathbb{X}$ = customer reviews
  - $\mathbb{C}$ = positive, negative
- Email routing and spam filtering
  - $\mathbb{X}$ = emails
  - $\mathbb{C}$ = spam, not-spam

Table of Contents

The Vector Space Models

- To teach machines to understand our languages, we need to design a representation which they can manipulate
- Vector space model (VSM) is an algebraic model for representing texts as vectors
  - Based on a series of statistical semantics hypothesis: takes event frequencies in corpora as clues to discover latent semantic
  - Derived vectors from a frequency matrix
    - The structure of the matrix relates to the scope of application of the vector space model[4]

Similarity of Documents: The Term-Document Matrix

Hypothesis (Bag-of-words Hypothesis)

The frequencies of words in a document tend to indicate the relevance of the document to a query[5].

If documents and queries have similar column vectors in a term-document matrix, then they tend to have similar meanings.

$$\begin{array}{ll} & \text{Documents} \\ & \begin{array}{cccc} d_1 & d_2 & \cdots & \end{array} \\ \text{Terms} \begin{array}{c} t_1 \\ t_2 \\ \vdots \\ \\ \end{array} & \begin{array}{|c|c|c|c|} \hline {fd_1}_{t_1} & {fd_2}_{t_1} & & \\ \hline {fd_1}_{t_2} & {fd_2}_{t_2} & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{array}$$

Similarity of Words: The Word-Context Matrix

Hypothesis (Distributional Hypothesis)

Words that occur in similar contexts tend to have similar meanings[6, 7].

If words have similar row vectors in a word-context matrix, then they tend to have similar meanings.

Contexts

		c_1	c_2	$\cdots$	
	w_1	$fc1_{w_1}$	$fc2_{w_1}$		
Words	w_2	$fc1_{w_2}$	$fc2_{w_2}$		
	$\vdots$				

Similarity of Relations: The Pair-Pattern Matrix

Hypothesis (Extended Distributional Hypothesis)

Patterns co-occurring with similar word-pairs tend to have similar meanings[8].

If patterns have similar column vectors in a pair-pattern matrix, then they tend to express similar semantic relations.

Hypothesis (Latent Relation Hypothesis)

Word-pairs co-occurring in similar patterns tend to have similar semantic relations[9].

If word-pairs have similar row vectors in a pair-pattern matrix, then they tend to have similar semantic relations.

Patterns

		p_1	p_2	$\cdots$	
	$(w_1^a{:}w_1^b)$	$fp1_{(w_1^a:w_1^b)}$	$fp2_{(w_1^a:w_1^b)}$		
Word-pairs	$(w_2^a{:}w_2^b)$	$fp1_{(w_2^a:w_2^b)}$	$fp2_{(w_2^a:w_2^b)}$		
	$\vdots$				

Table of Contents

Construction of Vector Space Models

- Linguistic Processing
  - Tokenization
  - Normalization
  - Annotation
- Mathematical Processing[10]
  - Building the frequency matrix
  - Weighting the elements
  - Dimensionality reduction
  - Comparing the similarities

Table of Contents

What are CopeOpi scores?

- Sentiment scores of Chinese characters/words[11]
  - Sentiment polarities: positive/negative
  - Strength of sentiment polarities
  - $+1$(positive) ~ -1(negative)
- The meaning of a Chinese word
  $= f($the meanings of its composite characters$)$
- The sentiment of a Chinese word
  $= f($sentiments of its composite characters$)$

How to compute CopeOpi scores?

- How to get the sentiment score of a Chinese character?
  - Assume that
    - Characters in a positive opinion word tend to be positive
    - Characters in a negative opinion word tend to be negative
- The observation probabilities of a character in positive and negative opinion words
  - NTUSD (NTU Sentiment Dictionary)[12] as seed words

The computation scheme of CopeOpi scores

Computation Scheme (CopeOpi Scores)

Given two corpora

- $\mathbb{W}_p = \{$Chinese positive opinion words$\}$
  - the vocabulary $\mathbb{V}_p = \{$unique characters in $\mathbb{W}_p\}$
- $\mathbb{W}_n = \{$Chinese negative opinion words$\}$
  - the vocabulary $\mathbb{V}_n = \{$unique characters in $\mathbb{W}_n\}$

For each character $c_i \in \mathbb{V}_p \cup \mathbb{V}_n$, we can compute its CopeOpi score $\mathcal{COP}_{c_i}$

The computation scheme of CopeOpi scores

Computation Scheme (CopeOpi Scores)

The CopeOpi score $\mathcal{COP}_{c_i}$ of a character c_i

$$\mathcal{P}_{c_i} = \frac{fp_{c_i} / \sum_{c \in \mathbb{V}_p} fp_c}{fp_{c_i} / \sum_{c \in \mathbb{V}_p} fp_c + fn_{c_i} / \sum_{c \in \mathbb{V}_n} fn_c}$$

$$\mathcal{N}_{c_i} = \frac{fn_{c_i} / \sum_{c \in \mathbb{V}_n} fn_c}{fp_{c_i} / \sum_{c \in \mathbb{V}_p} fp_c + fn_{c_i} / \sum_{c \in \mathbb{V}_n} fn_c}$$

$$\mathcal{COP}_{c_i} = \mathcal{P}_{c_i} - \mathcal{N}_{c_i}$$

	$\mathbb{W}_p$	$\mathbb{W}_n$
c_i	fp_{c_i}	fn_{c_i}

The computation scheme of CopeOpi scores

Computation Scheme (CopeOpi Scores)[13]

The CopeOpi score $\mathcal{COP}_w$ of a word $w = c_1 c_2 \cdots c_l$

$$\mathcal{COP}_{w=c_1 c_2 \cdots c_l} = \begin{cases} S_m(c_1 c_2 \cdots c_l) & \text{if the morphological type of } c_1 c_2 \cdots c_l \text{ is } m \\ \frac{1}{l} \sum_{j=1}^{l} \mathcal{COP}_{c_l} & \text{otherwise} \end{cases}$$

The applications of CopeOpi scores

- ANTUSD (Augmented NTU Sentiment Dictionary)[14]
  - A collection of opinion statistics in several annotation works
  - Each word in the dictionary is recorded with
    - The number of opinion annotations
    - The CopeOpi score

Table of Contents

Motivations for general CopeOpi scores

- In the character-context matrix of CopeOpi scores, the units are characters
  - Advantages: solves the coverage problem, since character types are much less than word types, scores of words can be computed if scores of characters are available
  - Disadvantages: can not be applied to languages whose characters have no meanings

	$\mathbb{W}_p$	$\mathbb{W}_n$
c_i	fp_{c_i}	fn_{c_i}

Motivations for general CopeOpi scores

- In the character-context matrix of CopeOpi scores, the contexts are opinion words
  - Advantages: reduces the noise of irrelevant words and ensures the precision of resulting scores
  - Disadvantages: limits the exploration of words excluded from seed words
- What other words shall we care about?
  - The domain-related opinion words
    - 浩然前廣場的草皮綠油油！
    - 最近的股市綠油油...
- Standard domain-independent sentiment lexicons are helpful but not sufficient for sentiment analysis

	$\mathbb{W}_p$	$\mathbb{W}_n$
c_i	fp_{c_i}	fn_{c_i}

Why can CopeOpi scores be generalized?

- The core of CopeOpi scores is a bag-of-characters method
  - A kind of statistical bag-of-units techniques
    - Commonly used in nature language processing (NLP)
    - Can be applied to different units of texts
- The premises of the formulas are that
    - Characters in a positive opinion word tend to be positive
    - Characters in a negative opinion word tend to be negative
  - Likewise, we can assume that
    - Words in a positive document tend to be positive
    - Words in a negative document tend to be negative
  - Moreover, we can assume that
    - Words in a documents of some categories tend to be in those categories

How to generalize CopeOpi scores?

- Change the basic units
  - Characters $\Rightarrow$ words
- Change the contexts
  - Chinese opinion words $\Rightarrow$ binary annotated documents

$$\begin{array}{c|c|c|} & \mathbb{W}_p & \mathbb{W}_n \\ \cline{2-3} c_i & fp_{c_i} & fn_{c_i} \\ \cline{2-3} & & \\ \cline{2-3} \end{array} \Rightarrow \begin{array}{c|c|c|} & \mathbb{D}_p & \mathbb{D}_n \\ \cline{2-3} w_i & fp_{w_i} & fn_{w_i} \\ \cline{2-3} & & \\ \cline{2-3} \end{array}$$

The computation scheme of general CopeOpi scores

Computation Scheme (General CopeOpi Scores)

Given two corpora

- $\mathbb{D}_p = \{\langle d, c\rangle \mid c = p\}$
  - the vocabulary $\mathbb{V}_p = \{$unique words in $\mathbb{D}_p\}$
- $\mathbb{D}_n = \{\langle d, c\rangle \mid c = \overline{p}\}$
  - the vocabulary $\mathbb{V}_n = \{$unique words in $\mathbb{D}_n\}$

For each unique word $w_i \in \mathbb{V}_p \cup \mathbb{V}_n$, we can compute its CopeOpi score $\mathcal{COP}_{w_i}$

The computation scheme of general CopeOpi scores

Computation Scheme (General CopeOpi Scores)

The CopeOpi score $\mathcal{COP}_{w_i}$ of a word w_i

$$\mathcal{P}_{w_i} = \frac{fp_{w_i} / \sum_{w \in \mathbb{V}_p} fp_w}{fp_{w_i} / \sum_{w \in \mathbb{V}_p} fp_w + fn_{w_i} / \sum_{w \in \mathbb{V}_n} fn_w}$$

$$\mathcal{N}_{w_i} = \frac{fn_{w_i} / \sum_{w \in \mathbb{V}_n} fn_w}{fp_{w_i} / \sum_{w \in \mathbb{V}_p} fp_w + fn_{w_i} / \sum_{w \in \mathbb{V}_n} fn_w}$$

$$\mathcal{COP}_{w_i} = \mathcal{P}_{w_i} - \mathcal{N}_{w_i}$$

	$\mathbb{D}_p$	$\mathbb{D}_n$
w_i	fp_{w_i}	fn_{w_i}

Confidence in general CopeOpi scores

- Zipf's law
  - Given some corpus of natural language, $f_w \propto 1/\mathrm{rank}(f_w)$
  - A few words that are very common
  - A very large number of words that are very rare
- Considering the latter case: rare words
  - Lack of sufficient statistics for precise scores
  - Easily biased and overestimated
    - A word w occurs in $\mathbb{D}_p$ once, $\mathcal{COP}_w = \frac{1/x-0}{1/x+0} = 1$
- Penalize rare words by a confidence value

Confidence in general CopeOpi scores

- We define
  - Rare words = words whose maximal class-frequency is less than the average class frequency of all words
  - The function of confidence values = a logistic function

$$fc_{w_i}^{\max} = \max(fc_{w_i}^1, fc_{w_i}^2, \ldots, fc_{w_i}^n)$$

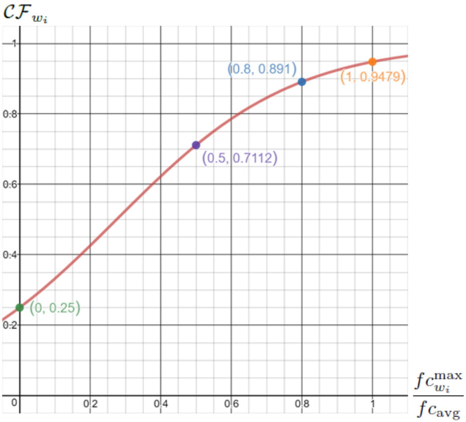
$$fc_{\text{avg}} = \frac{\sum_{j=1}^{m} \sum_{k=1}^{n} fc_{w_j}^k}{n \times m}$$

$$\mathcal{CF}_{w_i} = \begin{cases} 1 & \text{if } fc_{w_i}^{\max} \geq fc_{\text{avg}} \\ \dfrac{1}{1 + 3\exp^{-4(fc_{w_i}^{\max} / fc_{\text{avg}})}} & \text{otherwise} \end{cases}$$

$$\mathcal{CF}\text{-}\mathcal{COP}_{w_i} = \mathcal{CF}_{w_i} \times \mathcal{COP}_{w_i}$$

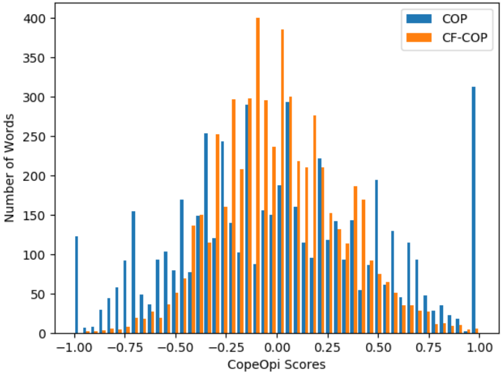
Confidence in general CopeOpi scores

Figure: The logistic function of $\mathcal{CF}$



Confidence in general CopeOpi scores

Figure: An example of $\mathcal{COP}$ and $\mathcal{CF}$-$\mathcal{COP}$



What are general CopeOpi Scores?

- Class-tendency scores of words
  - Class-tendencies: be in the class/not be in the class
  - Strength of class-tendencies
  - $+1$(be in the class) $\sim$ -1(not be in the class)
- Can be used in languages other than Chinese
  - Since we regard words as the basic units
- Can be used in binary text classification other than sentiment analysis
  - Since we take binary annotated documents as context

Table of Contents

Motivations for CopeOpi vectors

- There are many text classification problems with more than two categories
  - But CopeOpi scores can represent at most two classes by as positive or negative

How to construct CopeOpi vectors?

- How do we solve a multi-class classification problem?
  - Divide-and-conquer
    - Decomposing a multi-class classification problem into many binary classification sub-problems
- Decomposition strategies[15]
  - One-against-all strategy (OAA)
  - One-against-one strategy (OAO)

The computation scheme of CopeOpi vectors

Computation Scheme (CopeOpi Vectors)

Given n corpora $\mathbb{D}_{c_1}, \mathbb{D}_{c_2}, \ldots, \mathbb{D}_{c_n}$ and the corresponding classes $\mathbb{C} = \{c_1, c_2, \ldots, c_n\}$

- $\mathbb{D}_{c_i} = \{\langle d, c \rangle \mid c = c_i\}$
  - the vocabulary $\mathbb{V}_{c_i} = \{$unique words in $\mathbb{D}_{c_i}\}$

For each unique word $w_i \in \cup_{c \in \mathbb{C}} \mathbb{V}_c$, we can compute its CopeOpi vector $\overrightarrow{\mathcal{COP}}_{w_i}$ by one of the decomposition strategies.

The computation scheme of CopeOpi vectors

Computation Scheme (CopeOpi Vectors)(One-against-all)

For each class $c_j \in \mathbb{C}$, we can construct two opposite subests

- the positive set $\mathbb{P}_{w_i}^{c_j} = \{c_j\}$
  - the corpus $\mathbb{D}_{\mathbb{P}_{w_i}^{c_j}} = \{\mathbb{D}_{c_j}\}$
  - the vocabulary $\mathbb{V}_{\mathbb{P}_{w_i}^{c_j}} = \{\mathbb{V}_{c_j}\}$
- the negative set $\mathbb{N}_{w_i}^{c_j} = \mathbb{C} \setminus \{c_j\}$
  - the corpus $\mathbb{D}_{\mathbb{N}_{w_i}^{c_j}} = \{\mathbb{D}_c \mid c \in \mathbb{N}\}$
  - the vocabulary $\mathbb{V}_{\mathbb{N}_{w_i}^{c_j}} = \cup_{c \in \mathbb{N}_{w_i}^{c_j}} \mathbb{V}_c$

and compute its CopeOpi score $\mathcal{COP}_{w_i}^{c_j}$

The computation scheme of CopeOpi vectors

Computation Scheme (CopeOpi Vectors)(One-against-all)

The CopeOpi score $\mathcal{COP}^{c_j}_{w_i}$ of a word w_i with respect to class c_j

$$\mathcal{P}^{c_j}_{w_i} = \frac{fp^{c_j}_{w_i} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}}{}^{c_j}_{w_i}} fp^{c_j}_{w}}{fp^{c_j}_{w_i} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}}{}^{c_j}_{w_i}} fp^{c_j}_{w} + fn^{c_j}_{w_i} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}}{}^{c_j}_{w_i}} fn^{c_j}_{w}}$$

$$\mathcal{N}^{c_j}_{w_i} = \frac{fn^{c_j}_{w_i} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}}{}^{c_j}_{w_i}} fn^{c_j}_{w}}{fp^{c_j}_{w_i} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}}{}^{c_j}_{w_i}} fp^{c_j}_{w} + fn^{c_j}_{w_i} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}}{}^{c_j}_{w_i}} fn^{c_j}_{w}}$$

$$\mathcal{COP}^{c_j}_{w_i} = \mathcal{P}^{c_j}_{w_i} - \mathcal{N}^{c_j}_{w_i}$$

	$\mathbb{D}_{\mathbb{P}}{}^{c_j}_{w_i}$	$\mathbb{D}_{\mathbb{N}}{}^{c_j}_{w_i}$
w_i	$fp^{c_j}_{w_i}$	$fn^{c_j}_{w_i}$

The computation scheme of CopeOpi vectors

Computation Scheme (CopeOpi Vectors)(One-against-all)

The CopeOpi vector $\overrightarrow{\mathcal{COP}}_{w_i}$ of word w_i will be composed of these n CopeOpi scores.

$$\overrightarrow{\mathcal{COP}}_{w_i} = (\mathcal{COP}^{c_1}_{w_i}, \mathcal{COP}^{c_2}_{w_i}, \ldots, \mathcal{COP}^{c_n}_{w_i})$$

The computation scheme of CopeOpi vectors

Computation Scheme (CopeOpi Vectors)(One-against-one)

For each class-pair $c_j, c_k \in \mathbb{C}$, we can construct two opposite subests

- the positive set $\mathbb{P}_{w_i}^{c_{j,k}} = \{c_j\}$
  - the corpus $\mathbb{D}_{\mathbb{P}_{w_i}^{c_{j,k}}} = \{\mathbb{D}_{c_j}\}$
  - the vocabulary $\mathbb{V}_{\mathbb{P}_{w_i}^{c_{j,k}}} = \{\mathbb{V}_{c_j}\}$
- the negative set $\mathbb{N}_{w_i}^{c_{j,k}} = \{c_k\}$
  - the corpus $\mathbb{D}_{\mathbb{N}_{w_i}^{c_{j,k}}} = \{\mathbb{D}_{c_k}\}$
  - the vocabulary $\mathbb{V}_{\mathbb{N}_{w_i}^{c_{j,k}}} = \{\mathbb{V}_{c_k}\}$

and compute its CopeOpi score $\mathcal{COP}_{w_i}^{c_{j,k}}$

The computation scheme of CopeOpi vectors

Computation Scheme (CopeOpi Vectors)(One-against-one)

The CopeOpi score $\mathcal{COP}_{w_i}^{c_{j,k}}$ of a word w_i with respect to class-pair c_j, c_k

$$\mathcal{P}_{w_i}^{c_{j,k}} = \frac{fp_{w_i}^{c_{j,k}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}}{}_{w_i}^{c_{j,k}}} fp_w^{c_{j,k}}}{fp_{w_i}^{c_{j,k}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}}{}_{w_i}^{c_{j,k}}} fp_w^{c_{j,k}} + fn_{w_i}^{c_{j,k}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}}{}_{w_i}^{c_{j,k}}} fn_w^{c_{j,k}}}$$

$$\mathcal{N}_{w_i}^{c_{j,k}} = \frac{fn_{w_i}^{c_{j,k}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}}{}_{w_i}^{c_{j,k}}} fn_w^{c_{j,k}}}{fp_{w_i}^{c_{j,k}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}}{}_{w_i}^{c_{j,k}}} fp_w^{c_{j,k}} + fn_{w_i}^{c_{j,k}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}}{}_{w_i}^{c_{j,k}}} fn_w^{c_{j,k}}}$$

$$\mathcal{COP}_{w_i}^{c_{j,k}} = \mathcal{P}_{w_i}^{c_{j,k}} - \mathcal{N}_{w_i}^{c_{j,k}}$$

	$\mathbb{D}_{\mathbb{P}}{}_{w_i}^{c_{j,k}}$	$\mathbb{D}_{\mathbb{N}}{}_{w_i}^{c_{j,k}}$
w_i	$fp_{w_i}^{c_{j,k}}$	$fn_{w_i}^{c_{j,k}}$

The computation scheme of CopeOpi vectors

Computation Scheme (CopeOpi Vectors)(One-against-one)

The CopeOpi vector $\overrightarrow{\mathcal{COP}}_{w_i}$ of word w_i will be composed of these $\frac{1}{2}n(n-1)$ CopeOpi scores.

$$\overrightarrow{\mathcal{COP}}_{w_i} = (\mathcal{COP}_{w_i}^{c_{1,2}}, \mathcal{COP}_{w_i}^{c_{1,3}}, \ldots, \mathcal{COP}_{w_i}^{c_{n-1,n}})$$

Customized CopeOpi vectors

- OAA and OAO strategies guide the basic construction of CopeOpi vectors for multi-class text classification
- In general, any subset of classes can be grouped as a positive set or a negative set
  - $\mathbb{Q}$-against-$\mathbb{R}$ strategy

Customized CopeOpi vectors

Computation Scheme (CopeOpi Vectors)($\mathbb{Q}$-against-$\mathbb{R}$)

For any two subsets of classes $\mathbb{Q}, \mathbb{R}$, we can construct two opposite subsets

- the positive set $\mathbb{P}_{w_i}^{\mathbb{Q},\mathbb{R}} = \mathbb{Q}$
  - the corpus $\mathbb{D}_{\mathbb{P}_{w_i}^{\mathbb{Q},\mathbb{R}}} = \{\mathbb{D}_c \,|\, c \in \mathbb{Q}\}$
  - the vocabulary $\mathbb{V}_{\mathbb{P}_{w_i}^{\mathbb{Q},\mathbb{R}}} = \cup_{c \in \mathbb{Q}} \mathbb{V}_c$
- the negative set $\mathbb{N}_{w_i}^{\mathbb{Q},\mathbb{R}} = \mathbb{R}$
  - the corpus $\mathbb{D}_{\mathbb{N}_{w_i}^{\mathbb{Q},\mathbb{R}}} = \{\mathbb{D}_c \,|\, c \in \mathbb{R}\}$
  - the vocabulary $\mathbb{V}_{\mathbb{N}_{w_i}^{\mathbb{Q},\mathbb{R}}} = \cup_{c \in \mathbb{R}} \mathbb{V}_c$

and compute its CopeOpi score $\mathcal{COP}_{w_i}^{\mathbb{Q},\mathbb{R}}$

Customized CopeOpi vectors

Computation Scheme (CopeOpi Vectors)($\mathbb{Q}$-against-$\mathbb{R}$)

The CopeOpi score $\mathcal{COP}_{w_i}^{\mathbb{Q},\mathbb{R}}$ of a word w_i with respect to class subsets $\mathbb{Q}, \mathbb{R}$

$$\mathcal{P}_{w_i}^{\mathbb{Q},\mathbb{R}} = \frac{fp_{w_i}^{\mathbb{Q},\mathbb{R}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}w_i}^{\mathbb{Q},\mathbb{R}}} fp_w^{\mathbb{Q},\mathbb{R}}}{fp_{w_i}^{\mathbb{Q},\mathbb{R}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}w_i}^{\mathbb{Q},\mathbb{R}}} fp_w^{\mathbb{Q},\mathbb{R}} + fn_{w_i}^{\mathbb{Q},\mathbb{R}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}w_i}^{\mathbb{Q},\mathbb{R}}} fn_w^{\mathbb{Q},\mathbb{R}}}$$

$$\mathcal{N}_{w_i}^{\mathbb{Q},\mathbb{R}} = \frac{fn_{w_i}^{\mathbb{Q},\mathbb{R}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}w_i}^{\mathbb{Q},\mathbb{R}}} fn_w^{\mathbb{Q},\mathbb{R}}}{fp_{w_i}^{\mathbb{Q},\mathbb{R}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{P}w_i}^{\mathbb{Q},\mathbb{R}}} fp_w^{\mathbb{Q},\mathbb{R}} + fn_{w_i}^{\mathbb{Q},\mathbb{R}} \,/\, \sum_{w \in \mathbb{V}_{\mathbb{N}w_i}^{\mathbb{Q},\mathbb{R}}} fn_w^{\mathbb{Q},\mathbb{R}}}$$

$$\mathcal{COP}_{w_i}^{\mathbb{Q},\mathbb{R}} = \mathcal{P}_{w_i}^{\mathbb{Q},\mathbb{R}} - \mathcal{N}_{w_i}^{\mathbb{Q},\mathbb{R}}$$

	$\mathbb{D}_{\mathbb{P}w_i}^{\mathbb{Q},\mathbb{R}}$	$\mathbb{D}_{\mathbb{N}w_i}^{\mathbb{Q},\mathbb{R}}$
w_i	$fp_{w_i}^{\mathbb{Q},\mathbb{R}}$	$fn_{w_i}^{\mathbb{Q},\mathbb{R}}$

What are CopeOpi vectors?

- Word vectors, whose elements are classes-tendencies scores
  - Classes-tendencies: be in the classes/not be in the classes
  - Strength of classes-tendencies
  - $+1$(be in the classes) ~ -1(not be in the classes)
- Can be used in multi-class text classification

Experiments

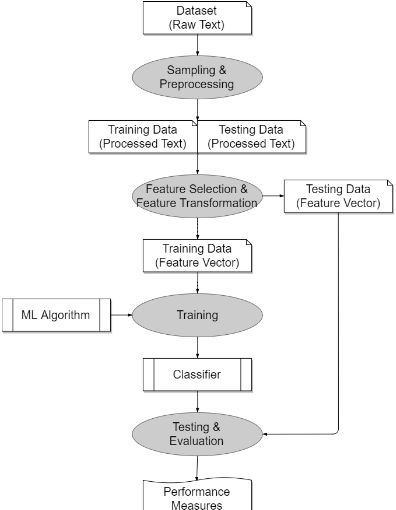
To verify the functionality of CopeOpi vectors, we make comparisons with several commonly used features of text, and examine these features on different types of classifiers to solve text classification problems

- Types
  - Sentiment analysis (SA)
  - Topic categorization (TC)
- Languages
  - English (EN)
  - Chinese (ZH)

Table of Contents

Flowchart

Figure: Flowchart of experiments



Sampling and Preprocessing

- Sampling
  - A training set
  - A testing set
- Preprocessing: unified procedure for each language
  - For English: tokenizing, stripping tags, stripping punctuations, stripping multiple whitespaces, stripping numeric, removing stopwords, stripping shorts and stemming
  - For Chinese: word segmentation and remove characters that are outside UTF-8 `[\u4E00-\u9FFF]`.

Feature Selection and Feature Transformation

- Term-document matrix models
  - BoW and its LSA-truncated version BoW(LSA)
    - Bag-of-word
  - TF-IDF and its LSA-truncated version TF-IDF(LSA)
    - $TF(w_i, d_j) = fd_{j_{w_i}} / \sum_{w \in d_j} fd_{j_{w_i}}$
      $IDF_{w_i} = \log \frac{|\mathbb{D}|}{|\{j: w_i \in d_j\}|}$
      $TF\text{-}IDF(w_i, d_j) = TF(w_i, d_j) \times IDF_{w_i}$
- Word-context matrix models
  - Word2vec[16] and its extension Doc2vec[17]
  - GolVe[18]
    - Neural language models

Training Classifiers

- k-nearest neighbor classifiers (kNN)
- Naive Bayes classifiers (NB)
  - Multinomial distribution: BoW, TF-IDF
  - Gaussian distribution: others
- Logistic regression classifiers (LR)
- Support vector machines (SVM)
  - Linear kernel
- Neural networks (NN)
  - One hidden layer with size 100

Testing and Evaluation

- Precision, recall, F1-scores for binary classification
  - $Presision_c = TP_c/(TP_c + FP_c)$
    $Recall_c = TP_c/(TP_c + FN_c)$
    $F1_c = (2 \times Presision_c \times Recall_c)/(Presision_c + Recall_c)$
- Macro-F1 for multi-class classification
  - $Macro\text{-}F1 = \frac{1}{|\mathbb{C}|}\sum_{c \in \mathbb{C}} F1_c$

Table: The contingency table of binary classification

		Real	
		True	False
Predicted	True	True positive (TP)	False positive (FP)
	False	False negative (FN)	True negative (TN)

Table of Contents

Datasets

- Both are 5-star integer ratings
  - $+5$(positive) $\sim$ $+1$(negative)

Table: Sentiment analysis datasets

Name	Language	Description	Source
Yelp Dataset[19]	English	Customer reviews about local business such as restaurants, hair stylists, mechanics, etc.	Yelp
MioChnCorp[20]	Chinese	Customer reviews about hotels.	Dianping

Experiments Datasets

- 15000 samples
- Train-test-split 0.5/0.5
- Balanced

Table: Sentiment analysis experiments datasets

<table>
<tr><th></th><th>Rating-1</th><th>Rating-2</th><th>Rating-3</th><th>Rating-4</th><th>Rating-5</th></tr>
<tr><td>SA(A)(2)</td><td colspan="2">Negative</td><td></td><td colspan="2">Positive</td></tr>
<tr><td>SA(B)(3)</td><td>Negative</td><td></td><td>Neutral</td><td></td><td>Positive</td></tr>
<tr><td>SA(C)(5)</td><td>Rating-1</td><td>Rating-2</td><td>Rating-3</td><td>Rating-4</td><td>Rating-5</td></tr>
</table>

Results and Observations 1

Figure: F1-score of SA(EN)(A)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi[1]	0.8246	0.8427	0.8439	0.8440	0.8441
BoW[3830]	0.3790	0.8637	0.8917	0.8969	0.8713
BoW(LSA)[100]	0.7434	0.7848	0.8436	0.8457	0.8308
TF-IDF[3830]	0.3481	0.8628	0.8997	0.8924	0.8689
TF-IDF(LSA)[100]	0.7691	0.7884	0.8781	0.8784	0.8715
Word2vec[160]	0.8091	0.7543	0.8801	0.8805	0.8744
GolVe[160]	0.8161	0.7563	0.8649	0.8755	0.8808
Doc2vec[10]	0.7863	0.8149	0.8228	0.8228	0.8215

Figure: F1-score of SA(ZH)(A)

Feature[size]	kNN	NB	LR	SVM	NN
ANTUSD[1]	0.7254	0.7257	0.7502	0.7491	0.7625
CopeOpi[1]	0.8603	0.8694	0.8698	0.8696	0.8735
BoW[3122]	0.8115	0.8790	0.8838	0.8912	0.8433
BoW(LSA)[100]	0.8150	0.8150	0.8648	0.8680	0.8551
TF-IDF[3122]	0.8325	0.8815	0.8928	0.8851	0.8371
TF-IDF(LSA)[100]	0.8332	0.8167	0.8812	0.8808	0.8709
Word2vec[160]	0.8576	0.8356	0.8898	0.8914	0.8812
GolVe[160]	0.8567	0.8373	0.8728	0.8775	0.8811
Doc2vec[10]	0.7167	0.7307	0.7548	0.7546	0.7584

Results and Observations 1

- SA(A)
  - Binary text classification
  - CopeOpi = general CopeOpi scores
- Compare the best F1-score of CopeOpi and the best F1-score of each experiment
  - Lose by 5.56% in SA(EN)
  - Lose by 1.93% in SA(ZH)
- This shows that the computation scheme of general CopeOpi scores is feasible

Results and Observations 2

Figure: F1-score of SA(ZH)(A)

Feature[size]	kNN	NB	LR	SVM	NN
ANTUSD[1]	0.7254	0.7257	0.7502	0.7491	0.7625
CopeOpi[1]	0.8603	0.8694	0.8698	0.8696	0.8735
BoW[3122]	0.8115	0.8790	0.8838	0.8912	0.8433
BoW(LSA)[100]	0.8150	0.8150	0.8648	0.8680	0.8551
TF-IDF[3122]	0.8325	0.8815	0.8928	0.8851	0.8371
TF-IDF(LSA)[100]	0.8332	0.8167	0.8812	0.8808	0.8709
Word2vec[160]	0.8576	0.8356	0.8898	0.8914	0.8812
GolVe[160]	0.8567	0.8373	0.8728	0.8775	0.8811
Doc2vec[10]	0.7167	0.7307	0.7548	0.7546	0.7584

Results and Observations 2

- SA(ZH)(A)
  - CopeOpi scores in ANTUSD
- Compare the F1-scores of CopeOpi and the F1-scores of ANTUSD
  - Outperform by more than 10%
- This shows general CopeOpi scores function normally without manually filtering non-opinion words and are more applicable to the dataset

Results and Observations 3

Figure: F1-score of SA(EN)(B)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[3]	0.7375	0.7533	0.7585	0.7571	0.7641
CopeOpi(OAO)[3]	0.7533	0.7487	0.7617	0.7628	0.7671
CopeOpi(OAA+OAO)[6]	0.7527	0.7503	0.7648	0.7673	0.7713
BoW[3859]	0.6014	0.7571	0.7857	0.7873	0.7473
BoW(LSA)[100]	0.6056	0.6738	0.7379	0.7383	0.7088
TF-IDF[3859]	0.5657	0.7531	0.7961	0.7793	0.7410
TF-IDF(LSA)[100]	0.6375	0.6797	0.7739	0.7719	0.7483
Word2vec[160]	0.6630	0.6273	0.7711	0.7728	0.7601
GolVe[160]	0.6915	0.6462	0.7625	0.7745	0.7799
Doc2vec[10]	0.6127	0.6460	0.6594	0.6601	0.6598

Figure: F1-score of SA(ZH)(B)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[3]	0.7378	0.7304	0.7664	0.7669	0.7682
CopeOpi(OAO)[3]	0.7522	0.7250	0.7654	0.7655	0.7691
CopeOpi(OAA+OAO)[6]	0.7507	0.7270	0.7686	0.7691	0.7694
BoW[3092]	0.6278	0.7683	0.7653	0.7640	0.7099
BoW(LSA)[100]	0.6426	0.6704	0.7362	0.7368	0.7094
TF-IDF[3092]	0.6306	0.7700	0.7736	0.7536	0.7063
TF-IDF(LSA)[100]	0.6745	0.7033	0.7555	0.7541	0.7351
Word2vec[160]	0.7101	0.7079	0.7668	0.7631	0.7382
GolVe[160]	0.6914	0.6943	0.7408	0.7475	0.7498
Doc2vec[10]	0.5304	0.5561	0.5854	0.5857	0.5854

Results and Observations 3

Figure: F1-score of SA(EN)(C)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[5]	0.4636	0.4790	0.4781	0.4761	0.4771
CopeOpi(OAO)[10]	0.4670	0.4733	0.4789	0.4794	0.4793
CopeOpi(OAA+OAO)[15]	0.4628	0.4752	0.4778	0.4800	0.4728
BoW[3919]	0.3257	0.4904	0.5018	0.4890	0.4624
BoW(LSA)[100]	0.3520	0.4146	0.4497	0.4412	0.4235
TF-IDF[3919]	0.3087	0.4867	0.5075	0.4770	0.4552
TF-IDF(LSA)[100]	0.3653	0.4300	0.4839	0.4820	0.4664
Word2vec[160]	0.3980	0.4013	0.4864	0.4741	0.4683
GolVe[160]	0.4028	0.4104	0.4761	0.4693	0.5023
Doc2vec[10]	0.3510	0.4030	0.3922	0.3800	0.4087

Figure: F1-score of SA(ZH)(C)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[5]	0.4561	0.4395	0.4757	0.4709	0.4765
CopeOpi(OAO)[10]	0.4489	0.4434	0.4706	0.4693	0.4773
CopeOpi(OAA+OAO)[15]	0.4521	0.4448	0.4723	0.4738	0.4718
BoW[3090]	0.3804	0.4964	0.4930	0.4767	0.4421
BoW(LSA)[100]	0.3730	0.4075	0.4611	0.4526	0.4346
TF-IDF[3090]	0.3867	0.4949	0.4936	0.4647	0.4337
TF-IDF(LSA)[100]	0.3960	0.4402	0.4796	0.4653	0.4531
Word2vec[160]	0.4241	0.4468	0.4899	0.4764	0.4650
GolVe[160]	0.4061	0.4288	0.4678	0.4558	0.4800
Doc2vec[20]	0.2838	0.3077	0.3515	0.3437	0.3527

Results and Observations 3

- SA(B), SA(C)
  - Multi-class text classification
  - CopeOpi = CopeOpi vectors
- Compare the F1-scores of CopeOpi and the F1-scores of each experiment
  - Lose by 2.49% in SA(EN)(B)
  - Lose by 2.79% in SA(EN)(C)
  - Lose by 0.42% in SA(ZH)(B)
  - Lose by 1.92% in SA(ZH)(C)
- This shows that the computation scheme of CopeOpi vectors is feasible

Results and Observations 4

- SA(A)
  - Binary text classification
  - CopeOpi = general CopeOpi scores
  - In 2/10 exps for each classifier, the F1-scores of CopeOpi are better than the average F1-score
- SA(B), SA(C)
  - Multi-class text classification
  - CopeOpi = CopeOpi vectors
  - In 20/20 exps for each classifier, the F1-scores of CopeOpi are better than the average F1-score of each classifier
- This shows that CopeOpi vectors in multi-class text classification is more effective then CopeOpi scores in binary text classification

Table of Contents

Datasets

- Both are 20 classes

Table: Topic categorization datasets

Name	Language	Train	Test	Total	Balanced
20Newgroup[21]	English	11314	7532	18846	Yes
Fudan Corpus	Chinese	9804	9833	19637	No

Experiments Datasets

- TC(EN) train-test-split 0.6/0.4
- TC(ZH) train-test-split 0.5/0.5

Table: Topic categorization experiments datasets

	Train	Test	Total	Balanced
TC(EN)(A)(20)	11314	7532	18846	True
TC(EN)(B)(7)	11314	7532	18846	False
TC(EN)(C)(5)	2936	1955	4891	True
TC(EN)(D)(4)	1952	1301	3253	True
TC(ZH)(A)(20)	9804	9833	19637	False
TC(ZH)(B)(9)	9318	9331	18649	False
TC(ZH)(C)(11)	486	502	988	True

Results and Observations 1a

Figure: F1-score of TC(EN)(A)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[20]	0.8063	0.7779	0.8111	0.8096	0.8095
CopeOpi(OAO)[190]	0.7760	0.7519	0.7913	0.7914	0.7834
CopeOpi(OAA+OAO)[210]	0.7831	0.7573	0.7944	0.7941	0.7830
BoW[8647]	0.4416	0.7969	0.7781	0.7486	0.8003
BoW(LSA)[100]	0.5153	0.5329	0.6365	0.6437	0.6609
TF-IDF[8647]	0.6511	0.7933	0.8011	0.8122	0.8220
TF-IDF(LSA)[100]	0.6654	0.6651	0.7315	0.7343	0.7478
Word2vec[160]	0.6696	0.6049	0.7272	0.7281	0.7285
GolVe[160]	0.6291	0.5654	0.6810	0.7006	0.7008
Doc2vec[20]	0.6219	0.6503	0.6592	0.6598	0.6707

Figure: F1-score of TC(ZH)(A)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[20]	0.6125	0.6150	0.5160	0.6464	0.6299
CopeOpi(OAO)[190]	0.6337	0.5962	0.5458	0.6347	0.6491
CopeOpi(OAA+OAO)[210]	0.6287	0.6003	0.5474	0.6422	0.6378
BoW[46409]	0.4157	0.5448	0.7742	0.7709	0.7974
BoW(LSA)[100]	0.4231	0.3929	0.3743	0.4421	0.5793
TF-IDF[46409]	0.6136	0.3138	0.4969	0.7848	0.7788
TF-IDF(LSA)[100]	0.6053	0.5624	0.4855	0.6322	0.7492
Word2vec[160]	0.5698	0.4255	0.4840	0.6517	0.7560
GolVe[160]	0.5666	0.4380	0.3703	0.5195	0.6459
Doc2vec[80]	0.6809	0.6471	0.5598	0.6706	0.6680

Results and Observations 1a

- TC(EN)(A), TC(ZH)(A)
  - Both corpora contain 20 categories
- Compare the best F1-score of CopeOpi and the best F1-score of each experiment
  - Lose by 1.10% in TC(EN)(A)
  - Lose by 14.83% in TC(ZH)(A)
- Considering
  - The results of SA shows that CopeOpi performs better in Chinese corpus than in English corpus
  - The preprocessing procedures are unified for each language
- The bad results of TC(ZH)(A) should caused by corpus itself rather than properties of language or preprocessing
  - Except languages, the biggest difference between their corpus is the balance
  - CopeOpi can not function well in unbalanced corpora ?

Results and Observations 1b

Figure: F1-score of TC(EN)(B)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[7]	0.8504	0.8390	0.8471	0.8492	0.8544
CopeOpi(OAO)[21]	0.8439	0.8184	0.8405	0.8477	0.8477
CopeOpi(OAA+OAO)[28]	0.8501	0.8256	0.8465	0.8501	0.8481
BoW[8647]	0.5509	0.8315	0.8440	0.8137	0.8417
BoW(LSA)[100]	0.6253	0.6009	0.7144	0.7277	0.7414
TF-IDF[8647]	0.7136	0.7045	0.8166	0.8606	0.8568
TF-IDF(LSA)[100]	0.7403	0.7374	0.7965	0.8024	0.8191
Word2vec[160]	0.7537	0.6358	0.7716	0.7826	0.8022
GolVe[160]	0.7366	0.6294	0.7188	0.7481	0.7706
Doc2vec[30]	0.6696	0.7053	0.7113	0.7163	0.7221

Figure: F1-score of TC(ZH)(A)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[20]	0.6125	0.6150	0.5160	0.6464	0.6299
CopeOpi(OAO)[190]	0.6337	0.5962	0.5458	0.6347	0.6491
CopeOpi(OAA+OAO)[210]	0.6287	0.6003	0.5474	0.6422	0.6378
BoW[46409]	0.4157	0.5448	0.7742	0.7709	0.7974
BoW(LSA)[100]	0.4231	0.3929	0.3743	0.4421	0.5793
TF-IDF[46409]	0.6136	0.3138	0.4969	0.7848	0.7788
TF-IDF(LSA)[100]	0.6053	0.5624	0.4855	0.6322	0.7492
Word2vec[160]	0.5698	0.4255	0.4840	0.6517	0.7560
GolVe[160]	0.5666	0.4380	0.3703	0.5195	0.6459
Doc2vec[80]	0.6809	0.6471	0.5598	0.6706	0.6680

Results and Observations 1b

- TC(EN)(B), TC(ZH)(A)
  - Both corpora are unbalanced
- Compare the best F1-score of CopeOpi and the best F1-score of each experiment
  - Lose by 0.62% in TC(EN)(B)
  - Lose by 14.83% in TC(ZH)(A)
- CopeOpi functions as well as usual in one of them
  - Except languages, the biggest difference between their corpus is that some of the categories of TC(ZH)(A) have only a few samples
- We deduce that The bad results of TC(ZH)(A) should caused by the small-sized categories, not the unbalanced corpus
  - CopeOpi can not function well in corpora with small-sized categories

Results and Observations 2

Figure: F1-score of TC(ZH)(B)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[9]	0.9121	0.8270	0.8955	0.9120	0.9126
CopeOpi(OAO)[36]	0.9013	0.7824	0.8722	0.9001	0.8985
CopeOpi(OAA+OAO)[45]	0.9053	0.7965	0.8843	0.9091	0.9055
BoW[45992]	0.7218	0.8811	0.9293	0.9185	0.9356
BoW(LSA)[100]	0.8202	0.6767	0.8439	0.8715	0.8948
TF-IDF[45992]	0.8620	0.7141	0.9183	0.9400	0.9439
TF-IDF(LSA)[100]	0.8968	0.8284	0.9036	0.9142	0.9290
Word2vec[160]	0.8655	0.6691	0.8534	0.8938	0.9140
GolVe[160]	0.8771	0.7188	0.8183	0.8835	0.9093
Doc2vec[80]	0.9082	0.8947	0.9073	0.9079	0.9022

Figure: F1-score of TC(ZH)(C)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[11]	0.7850	0.7282	0.7722	0.8118	0.7977
CopeOpi(OAO)[55]	0.7377	0.7080	0.7355	0.7545	0.7544
CopeOpi(OAA+OAO)[66]	0.7551	0.7246	0.7370	0.7763	0.7809
BoW[23920]	0.3456	0.7269	0.8103	0.8174	0.8321
BoW(LSA)[100]	0.5992	0.5588	0.6507	0.7713	0.7752
TF-IDF[23920]	0.7296	0.4734	0.7398	0.9014	0.8706
TF-IDF(LSA)[100]	0.7206	0.6632	0.8518	0.8863	0.8934
Word2vec[160]	0.6248	0.6257	0.6574	0.7533	0.7507
GolVe[160]	0.4676	0.4769	0.3363	0.5830	0.5341
Doc2vec[70]	0.7529	0.7206	0.7459	0.7642	0.7747

Results and Observations 2

- TC(ZH)(B), TC(ZH)(C)
  - The corpus of TC(ZH)(C) is constituted by the corpus of the small-sized categories
  - The corpus of TC(ZH)(B) is constituted by the corpus of the rest categories
- Compare the best F1-score of CopeOpi and the best F1-score of each experiment
  - Lose by 3.12% in TC(ZH)(B)
  - Lose by 8.95% in TC(ZH)(C)
- This confirms that CopeOpi can not function well in corpora with small-sized categories

Results and Observations 3

Figure: F1-score of TC(EN)(C)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[5]	0.7679	0.7745	0.7727	0.7754	0.7777
CopeOpi(OAO)[15]	0.7465	0.7497	0.7558	0.7564	0.7502
CopeOpi(OAA+OAO)[20]	0.7593	0.7605	0.7662	0.7708	0.7702
BoW[8284]	0.4802	0.7582	0.7670	0.7470	0.7784
BoW(LSA)[100]	0.6067	0.6316	0.7221	0.7311	0.7393
TF-IDF[8284]	0.6512	0.7827	0.7987	0.7935	0.7842
TF-IDF(LSA)[100]	0.6644	0.6787	0.7663	0.7655	0.7762
Word2vec[160]	0.6717	0.6701	0.7352	0.7244	0.7307
GolVe[160]	0.6217	0.6459	0.7055	0.7131	0.7055
Doc2vec[20]	0.5802	0.6200	0.6376	0.6362	0.6358

Figure: F1-score of TC(EN)(D)

Feature[size]	kNN	NB	LR	SVM	NN
CopeOpi(OAA)[4]	0.8431	0.8414	0.8467	0.8434	0.8484
CopeOpi(OAO)[6]	0.8402	0.8376	0.8354	0.8371	0.8375
CopeOpi(OAA+OAO)[10]	0.8409	0.8414	0.8400	0.8425	0.8473
BoW[11851]	0.5755	0.8361	0.8164	0.8045	0.8456
BoW(LSA)[100]	0.7277	0.7302	0.7817	0.7785	0.7648
TF-IDF[11851]	0.8368	0.8306	0.8465	0.8549	0.8546
TF-IDF(LSA)[100]	0.8046	0.7867	0.8163	0.8276	0.8196
Word2vec[160]	0.7872	0.7681	0.8099	0.8055	0.7906
GolVe[160]	0.7092	0.7180	0.7673	0.7666	0.7537
Doc2vec[20]	0.7293	0.7481	0.7697	0.7612	0.7604

Results and Observations 3

- TC(EN)(C), TC(EN)(D)
  - Both corpora are constituted by the corpus with similar categories
- Compare the best F1-score of CopeOpi and the best F1-score of each experiment
  - Lose by 2.11% in TC(EN)(C)
  - Lose by 0.66% in TC(EN)(D)
- This shows that CopeOpi can function well even if the categories are similar

Table of Contents

Summary 1

- CopeOpi can produce comparable results with a smaller vector size and shorter training time
  - In 53/55 exps for each classier, the best F1-score of CopeOpi vectors in multi-class is better than the average F1-score
  - In 63/65 exps for each classier, the training time of CopeOpi is the shortest
  - In all exps, the vector size of CopeOpi(OAA) is the smallest

Summary 2

- Compared to the other features, CopeOpi provides stabler results than the others when applied to different types of classifiers
  - There are some results deviating, but in those cases the deviations are general phenomena for most of features

Summary 3

- Compared to the winners in multi-class text classification
  - Either BoW or TF-IDF
  - Eexcept the experiments whose corpus has small-sized categories, the difference of the best F1-score of CopeOpi and their F1-score of each experiments is at most 3.12%
- However, the training processes of BoW and TF-IDF cost most in terms of memory space and times
  - In 64/65 exps for each classier, the best F1-score of CopeOpi is better than the F1-score of BoW(LSA)[100]
  - In 50/65 exps for each classier, the best F1-score of CopeOpi is better than the F1-score of TF-IDF(LSA)[100]
- Since the vector sizes of BoW and TF-IDF are related to the number of vocabularies in corpora, in the cases with large corpora, CopeOpi will have advantages in its efficiency.

Table of Contents

Conclusions

- We propose a vector space model, the word vectors—CopeOpi vectors
  - From CopeOpi scores used in Chinese sentiment analysis
  - To CopeOpi vectors which can be used in multi-class text classification without being limited to languages
- We verify the effectiveness and efficiency of CopeOpi vectors by making comparisons with several commonly-used features for text classification
  - Various text classification problems in both English and Chinese
  - The results show that CopeOpi can produce comparable results with a smaller vector size and shorter training time
- In general, CopeOpi vectors are effective and efficient features for multi-class text classification

Table of Contents

Future Works

1. More careful term-weighting schemes
   - The original CopeOpi scores are computed from dictionaries
     - The term-weighting scheme of the current formula $fc \,/\, \sum_w fc_w$
   - But now we compute CopeOpi from nature language corpora
     - There may be a lot of unrelated words
     - CopeOpi needs a more careful term-weighting scheme
2. Strategies to customize CopeOpi vectors
   - The number of classes-pairs are exponential to the number of class
     - Flexibility
     - Difficulty

References I

[1] R. Feldman and J. Sanger, *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*.
New York, NY, USA: Cambridge University Press, 2006.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*.
New York, NY, USA: Cambridge University Press, 2008.

[3] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, pp. 1–47, Mar. 2002.

[4] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *J. Artif. Int. Res.*, vol. 37, pp. 141–188, Jan. 2010.

References II

[5] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, Nov. 1975.

[6] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[7] J. Firth, *A Synopsis of Linguistic Theory, 1930-1955*. 1957.

[8] D. Lin and P. Pantel, "Dirt @sbt@discovery of inference rules from text," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, (New York, NY, USA), pp. 323–328, ACM, 2001.

References III

[9] P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Trans. Inf. Syst.*, vol. 21, pp. 315–346, Oct. 2003.

[10] W. Lowe, "Towards a theory of semantic space," in *Proceedings of the Cognitive Science Society*, vol. 23, 2001.

[11] L.-w. Ku, Y.-s. Lo, and H.-h. Chen, "Using polarity scores of words for sentence-level opinion extraction," in *Proceedings of NTCIR-6 workshop meeting*, pp. 316–322, 2007.

[12] L.-W. Ku and H.-H. Chen, "Mining opinions from the web: Beyond relevance retrieval," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 12, pp. 1838–1850, 2007.

References IV

[13] L.-W. Ku, T.-H. Huang, and H.-H. Chen, "Using morphological and syntactic structures for chinese opinion analysis," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1260–1269, Association for Computational Linguistics, 2009.

[14] S.-M. Wang and L.-W. Ku, "Antusd: A large chinese sentiment dictionary," in *the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 2697–2702, 2016.

[15] M. Aly, "Survey on multiclass classification methods," *Neural Netw*, vol. 19, 2005.

References V

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

[17] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196, 2014.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[19] "Yelp dataset challenge." `https://www.yelp.com/dataset_challenge`, 2017.

References VI

[20] Y. Lin, H. Lei, J. Wu, and X. Li, "An empirical study on sentiment classification of chinese review using word embedding," *arXiv preprint arXiv:1511.01665*, 2015.

[21] "20 newsgroups." http://qwone.com/~jason/20Newsgroups, 2017.