Scalable Term Selection for Text Categorization

Jingyang Li, Maosong Sun
Tsinghua University, Beijing, China

Abstract: By involving the target dimensionality as a parameter, a new term selection criterion is constructed via controlling the average vector length, whose expected value can be given by an empirical formula.
# Table of Contents

1. Introduction
   1.1. Text Categorization
   1.2. Term Selection
2. Experiment Settings
3. Average Vector Length (AVL)
4. Scalable Term Selection (STS)
   4.1. Measuring Coverage and Discriminability
   4.2. The Combined Criterion
5. Experiments and the Final Algorithm
6. Further Observation and Discussion
1. Introduction

1.1. Text Categorization

Common TC phases:
- document vectorization
- dimensionality reduction (term selection)
- classifier learning
- classification and evaluation

Related definition:
- \( D \) — the training document set
- \( d_j \) — a document in \( D \)
- \( T \) — the term set
- \( t_i \) — a term in \( T \)
1.2. Term Selection

Term selection is necessary for

- removing \textit{irrelevant} terms and \textit{redundant} terms
- considerations on computational cost

Why not general feature selection techniques?

- Domain specific (\textit{ad hoc}) ones are always better.
- TC has the special \textit{sparserness* problem} (which is supposed to cause the low performance at low dimensionalities).

The current technique ($\chi^2$, IG, BNS, ...):

- a single criterion $\circ(t_i) \rightarrow$ a single rank list
- implicitly considered the \textit{discriminability} and the \textit{coverage} of $t_i$
  (correspond to the \textit{specificity} and the \textit{exhaustivity} of the term set $T$).
2. Experiment Settings

Document collections:

<table>
<thead>
<tr>
<th></th>
<th>CE</th>
<th>20NG</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Chinese</td>
<td>English</td>
<td></td>
</tr>
<tr>
<td>Num of categories</td>
<td>55</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Num of documents</td>
<td>71,674</td>
<td>18,821</td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>9 : 1</td>
<td>6 : 4</td>
<td>#training : #test</td>
</tr>
<tr>
<td>Num of terms,</td>
<td>1,067,717</td>
<td>30,220</td>
<td>( df(t_i) \geq 2 ) in the training set</td>
</tr>
</tbody>
</table>

Other settings:

Term selection: \( \chi^2 \) (baseline), STS

Term weighting: \( tfidf(t_i, d_j) = \log(tf(t_i, d_j) + 1) \cdot \log \left( \frac{df(t_i)+1}{Nd} \right) \)

Classifier: linear kernel SVMs

Evaluation: \( F_1 \)-measure
3. Average Vector Length (AVL)

Observation:

▶ Sparseness is the main reason of the low performance at low dimensionalities.
Section 3: Average Vector Length (AVL)

The basic idea:

- **vector length**: measures the sparseness of a document
- **average vector length**: measures the “sparseness of a term set”

Estimating AVL by training set:

- \(|d_j|\) — number of different terms in \(T\) contained by \(d_j\).
  - More plausible:
    \[
    \exp\left(\frac{\sum \log |d_j|}{|D|}\right)
    \]
  - Faster to compute (while \(T\) varies):
    \[
    \frac{\sum |d_j|}{|D|} = \frac{\sum df(t_i)}{|D|}
    \]

▷ So we define

\[
AVL_T = \frac{\sum_{t_i \in T} df(t_i)}{|D|}
\]
4. Scalable Term Selection (STS)

The basic idea: STS should automatically accommodate its favor of high coverage to different target dimensionalities.

4.1. Measuring Coverage and Discriminability

The metrics of coverage and discriminability should:

- not be highly positively correlated
- have a slight negative correlation, intuitively

coverage:

\[ \log df(t_i) \]
Section 4: Scalable Term Selection (STS)

**discriminability:**

**probability ratio:**

\[ PR(t_i, c) = \frac{P(t_i|c_+)}{P(t_i|c_-)} = \frac{df(t_i, c_+)/df(c_+)}{df(t_i, c_-)/df(c_-)} \]

\[ \log PR_{\text{max}}(t_i) = \log \max_c \{PR(t_i, c)\}, \text{ in which } c \text{ is a category.} \]
4.2. The Combined Criterion

\[ \zeta(t_i; \lambda) = \left( \frac{\lambda}{\log(\text{PR}(t_i))} + \frac{1 - \lambda}{\log(\text{df}(t_i))} \right)^{-1}, \quad \lambda \in [0, 1] \]

The optimal \( \lambda \) is a function of the target dimensionality \( k \):

\[ \lambda^*(k) = \arg \max_\lambda F_1(k) \]

There is a corresponding optimal AVL:

\[ AVL^*(k) \leftrightarrow \lambda^*(k) \]

Is there a empirical formula to estimate \( \lambda^* \) or \( AVL^* \)?

\[ AVL^\circ(k) \doteq AVL^*(k) \]
5. Experiments and the Final Algorithm

1. For each dimensionality $k$, we manually search $AVL^*(k)$ in integers.
2. For each $AVL$, search the corresponding $\lambda$;
   $AVL(\lambda; k)$ is monotone and fast to compute.
Section 5: Experiments and the Final Algorithm

CE

20NG

![Graphs showing AVL' and dimensionality (k) for CE and 20NG datasets with λ^∗ and dimensionality (k) graphs.]
Observations:

- $F_1(k)$ is not very sensitive to small $\Delta AVL(k)$ near $AVL^*(k)$.
- $\gamma = \frac{\log(AVL^*(k))/\log(AVL_T)}{\log(k)} \approx 0.085$

$(AVL_{T_{CE}} = 898.53, AVL_{T_{20NG}} = 82.16)$
Final formulas:

The empirical estimation of $AVL^*(k)$ is

$$AVL^*(k) = \exp(\gamma \log(AVL_T) \cdot \log(k))$$
$$= AVL_T^{\gamma \log(k)}$$

and the final STS criterion is

$$\zeta(t_i, k) = \zeta(t_i; \lambda(AVL^*(k)))$$
$$= \zeta \left( t_i; \lambda \left( AVL_T^{\gamma \log(k)} \right) \right)$$

in which $\lambda(AVL)$ is still computed by search.
6. Further Observation and Discussion

Comparing the selection results of STS and $\chi^2$:

- For most $\lambda$, the *Spearman’s rank correlation coefficient* between $\eta(t_i; \lambda)$ and $\chi^2(t_i)$ is bigger than 0.999.
- Selection areas on *(coverage, discriminability)* of CE and 20NG:
About sparseness:

- *collection sparseness*: too few training samples. This backroom factor might lead to the different behaviors of STS on CE and 20NG.
- *document sparseness*: this study.

Adaptability:

- Performance
- Computational cost: depends on AVL or $k$

plots