Probabilistic Semantic Similarity Measurements for Noisy Short Texts Using Wikipedia Entities

Masumi Shirakawa¹, Kotaro Nakayama², Takahiro Hara¹, Shojiro Nishio¹

¹Osaka University, Osaka, Japan
²University of Tokyo, Tokyo, Japan
Challenge in short text analysis

Statistics are not always enough.

A year and a half after Google pulled its popular search engine out of mainland China

Baidu and Microsoft did not disclose terms of the agreement
Statistics are not always enough.

A year and a half after Google pulled its popular search engine out of mainland China

Baidu and Microsoft did not disclose terms of the agreement

They are talking about...

Search engines and China
Challenge in short text analysis

Statistics are not always enough.

A year and a half after Google pulled its popular search engine out of mainland China

Baidu and Microsoft did not disclose terms of the agreement

How do machines know that the two sentences mention about the similar topic?

They are talking about...

Search engines and China
A year and a half after Google pulled its popular search engine out of mainland China, Baidu and Microsoft did not disclose terms of the agreement.
Related work

ESA: Explicit Semantic Analysis [Gabrilovich07]
Add Wikipedia articles (entities) to a text as its semantic representation.

1. Get search ranking of Wikipedia for each term (i.e. Wiki articles and scores).
2. Simply sum up the scores for aggregation.

Input: \( T \)

Key term extraction

Related entity finding

Aggregation

Output: ranked list of \( c \)
Problems in real world noisy short texts

“Noisy” means semantically noisy in this work.
(We do not handle informal or casual surface forms, or misspells)

Term ambiguity
• Apple (fruit) should not be related with Microsoft.

Fluctuation of term dominance
• A term is not always important in texts.

We explore more effective aggregation method.
We propose Extended naïve Bayes to aggregate related entities.
When input is multiple terms

Apply naïve Bayes [Song11] to multiple terms $t_1, ..., t_K$ to obtain related entity $c$ using each probability $P(c|t_k)$. 

$$P(c|t_1, ..., t_K) = \frac{P(t_1, ..., t_K|c)P(c)}{P(t_1, ..., t_K)} = \frac{P(c) \prod_k P(t_k|c)}{P(t_1, ..., t_K)} = \prod_k \frac{P(c|t_k)}{P(c)^{K-1}}$$

Diagram:
- $t_1$ Apple
- $t_2$ product
- $t_K$ new
- $c =$ "iPhone"
- Compute $P(c|t_1, ..., t_K)$ for each related entity $c$
When input is multiple terms

Apply naïve Bayes [Song11] to multiple terms $t_1, \ldots, t_K$ to obtain related entity $c$ using each probability $P(c|t_k)$.

\[
P(c|t_1, \ldots, t_K) = \frac{P(t_1, \ldots, t_K|c)P(c)}{P(t_1, \ldots, t_K)} = \frac{P(c) \prod_k P(t_k|c)}{P(t_1, \ldots, t_K)} = \frac{\prod_k P(c|t_k)}{P(c)^{K-1}}
\]

By using naïve Bayes, entities that are related to multiple terms can be boosted.
When input is text

Not “multiple terms” but “text,” i.e., we don’t know which terms are key terms.

We developed extended naïve Bayes to solve this problem.

\[
P(c|t_1) \quad P(c|t_2) \quad P(c|t_K)
\]

\[c = “iPhone”\]
Extended naïve Bayes

\[ T' = \{t_1\} \]
\[ T' = \{t_1, t_2\} \]
\[ T' = \{t_1, \ldots, t_K\} \]

Candidates of key term

Apply naïve Bayes to each state \( T' \)

Probability that the set of key terms \( T \) is a state \( T' \): \( P(T = T') \)

<table>
<thead>
<tr>
<th>Apple</th>
<th>product</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td></td>
</tr>
</tbody>
</table>
Extended naïve Bayes

\[ T' = \{ t_1 \} \]

\[ T' = \{ t_1, t_2 \} \]

\[ T' = \{ t_1, \cdots, t_K \} \]

Apply naïve Bayes to each state \( T' \)

\[ \sum_{T'} P(c|T') P(T = T') = \frac{\prod_k (P(t_k \in T) P(c|t_k) + (1 - P(t_k \in T)) P(c))}{P(c)^{K-1}} \]
Extended naïve Bayes

Term dominance is incorporated into naïve Bayes

\[
\sum_{T'} P(c|T') P(T = T') = \frac{\prod_{k} (P(t_k \in T)P(c|t_k) + (1 - P(t_k \in T))P(c))}{P(c)^{K-1}}
\]
Experiments on short text sim datasets

[Datasets] Four datasets derived from word similarity datasets using dictionary

[Comparative methods] Original ESA [Gabrilovich07], ESA with 16 parameter settings

[Metrics] Spearman’s rank correlation coefficient

<table>
<thead>
<tr>
<th>Method</th>
<th>Pilot</th>
<th>MC</th>
<th>RG</th>
<th>WS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>0.733</td>
<td>0.777</td>
<td>0.681</td>
<td>0.506</td>
</tr>
<tr>
<td>KEY-A-L (ESA-same)</td>
<td>0.824</td>
<td>0.826</td>
<td>0.727</td>
<td>0.542</td>
</tr>
<tr>
<td>KEY-A-L-COS</td>
<td>0.823</td>
<td>0.754</td>
<td>0.690</td>
<td>0.571</td>
</tr>
<tr>
<td>KEY-A-logL</td>
<td>0.797</td>
<td>0.814</td>
<td>0.710</td>
<td>0.559</td>
</tr>
<tr>
<td>KEY-A-logL-COS</td>
<td>0.771</td>
<td>0.814</td>
<td>0.626</td>
<td>0.447</td>
</tr>
<tr>
<td>KEY-logA-L</td>
<td>0.820</td>
<td>0.856</td>
<td>0.650</td>
<td>0.528</td>
</tr>
<tr>
<td>KEY-logA-logL</td>
<td>0.866</td>
<td>0.840</td>
<td>0.713</td>
<td>0.505</td>
</tr>
<tr>
<td>KEY-logA-logL-COS</td>
<td>0.785</td>
<td>0.866</td>
<td>0.706</td>
<td>0.553</td>
</tr>
<tr>
<td>IDF-A-L</td>
<td>0.737</td>
<td>0.893</td>
<td>0.790</td>
<td>0.392</td>
</tr>
<tr>
<td>IDF-A-L-COS</td>
<td>0.886</td>
<td>0.835</td>
<td>0.791</td>
<td>0.523</td>
</tr>
<tr>
<td>IDF-A-logL</td>
<td>0.845</td>
<td>0.869</td>
<td>0.778</td>
<td>0.509</td>
</tr>
<tr>
<td>IDF-A-logL-COS (ESA-adjusted)</td>
<td>0.885</td>
<td>0.894</td>
<td>0.806</td>
<td>0.569</td>
</tr>
<tr>
<td>Original ESA</td>
<td>0.797</td>
<td>0.833</td>
<td>0.698</td>
<td>0.562</td>
</tr>
<tr>
<td>Our method</td>
<td>0.857</td>
<td>0.840</td>
<td>0.717</td>
<td>0.573</td>
</tr>
</tbody>
</table>

ESA with well-adjusted parameter is superior to our method for “clean” texts.
Tweet clustering

K-means clustering using the vector of related entities for measuring distance

**Dataset** 12,385 tweets including 13 topics

<table>
<thead>
<tr>
<th>hashtag</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>#MacBook</td>
<td>1,251</td>
</tr>
<tr>
<td>#MySQL</td>
<td>1,241</td>
</tr>
<tr>
<td>#NFL</td>
<td>1,044</td>
</tr>
<tr>
<td>#MLB</td>
<td>752</td>
</tr>
<tr>
<td>#NASCAR</td>
<td>878</td>
</tr>
<tr>
<td>#Silverlight</td>
<td>221</td>
</tr>
<tr>
<td>#Ubuntu</td>
<td>988</td>
</tr>
<tr>
<td>#NHL</td>
<td>1,045</td>
</tr>
<tr>
<td>#MLS</td>
<td>981</td>
</tr>
<tr>
<td>#VMWare</td>
<td>890</td>
</tr>
<tr>
<td>#Chrome</td>
<td>1,018</td>
</tr>
<tr>
<td>#NBA</td>
<td>1,085</td>
</tr>
<tr>
<td>#UFC</td>
<td>991</td>
</tr>
</tbody>
</table>

**Comparative methods** Bag-of-words (BOW), ESA with the same parameter, ESA with well-adjusted parameter

**Metric** Average of Normalized Mutual Information (NMI), 20 runs
Results

![Graph showing NMI score vs. number of related entities for BOW, ESA-same, ESA-adjusted, and Our method. The graph includes data points for 0.429, 0.524, and 0.567 at different data points. A note indicates a p-value < 0.01.]

- BOW
- ESA-same
- ESA-adjusted
- Our method

p-value < 0.01
Results

Our method outperformed ESA with well-adjusted parameter for noisy short texts.

p-value < 0.01
Conclusion

We proposed extended naïve Bayes to derive related Wikipedia entities given a real world noisy short text.

[Future work]
Tackle multilingual short texts
Develop applications of the method