Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge

Xia Hu,¹,² Nan Sun,¹ Chao Zhang,¹ Tat-Seng Chua¹

¹School of Computing
National University of Singapore

²School of Computer Science and Engineering
BeiHang University

November 2, 2009
1 Introduction

2 Proposed Framework

3 Evaluation

4 Conclusion and Future Work
Aggregated Search

- The form of browsing search results.

Improving the Clustering of Short Texts

Xia Hu

Outline

Introduction

Proposed Framework

Evaluation

Conclusion and Future Work

Clustering

- Aggregated Search

The form of browsing search results.

- The Dark Knight (2008)
  - Batman, Gordon and Harvey Dent are forced to deal with the chaos unleashed by an anarchist movement led by the maniacal Joker.
  - www.imdb.com/title/tt0470217

2. The Dark Knight (2008)
  - Batman - Movie Trailers - The Dark Knight
  - The follow-up to the action hit "The Dark Knight Rises," this movie continues the story of Bruce Wayne/Batman.
  - www.imdb.com/title/tt0470217

3. The Dark Knight (2008)
  - Wikipedia, the free encyclopedia
  - The Dark Knight Rises is a 2012 superhero film directed and co-written by Christopher Nolan. Based on the DC Comics character:
  - www.wikipedia.org/wiki/The_Dark_Knight_Rises

4. The Dark Knight Rises
  - A site devoted to Batman, the Dark Knight Detective. Very interactive site with loads of information about the comics world:
  - www.darkknight.com

5. The Dark Knight Rises
  - www.theofficialdarkknight.movie

6. The Dark Knight Rises Movie Reviews, Pictures - Rotten Tomatoes
  - The Dark Knight movie reviews, trailers - Check out Rotten Tomatoes The Dark Knight clips, pictures, and user reviews:
  - www.rottentomatoes.com/m/The_Dark_Knight_Rises

Conclusion and Future Work
Short Texts

- Short texts, such as the snippets, product descriptions, QA passages and image captions etc., have played important roles in current Web and IR applications.
- Unlike standard texts with lots of words in length, short texts, which only consist of a few phrases or 2–3 sentences, especially present great challenges in clustering.
- Problems: “data sparseness” & “semantic gap”.
Many methods have been proposed to improve the representation of standard text for clustering and classification, including “surface representation” [3,19] and “integrating world knowledge” [14].

Several clustering techniques were employed to place the search engine snippets to their highly relevant topic-coherent groups [5,29].

World knowledge bases have been found useful in improving the short text representation [1,23].
The General Framework

Hierarchical Resolution

Feature Generation

Feature Selection

Introduction

Proposed Framework

Evaluation

Conclusion and Future Work

Improve the Clustering of Short Texts

Xia Hu

Outline

Introduction

Proposed Framework

Evaluation

Conclusion and Future Work

Fig: Framework for feature constructor
“Jul 18, 2008 … It is the best American film of the year so far and likely to remain that way. Christopher Nolan’s The Dark Knight is revelatory, visceral …”

Fig: Syntax tree of the snippet
Original Feature Extraction

- Segment-level features.
- Phrase-level features.
  - *Sentence1*: [NP July 18 2008]
  - *Sentence2*: [NP It] [VP is] [NP the best American film] [PP of] [NP the year] [ADVP so far] and/CC [ADJP likely] [VP to remain] [NP that way]
  - *Sentence3*: [NP Christopher Nolan ‘s] [NP The Dark Knight] [VP is] [NP revelatory visceral]
- Word-level features.
Feature Generation

Two steps:

- the construction of basic features
  - seed phrases from internal semantics.
- the generation of external features.
  - external features from world knowledge bases.
There are redundancies between phrase level features and segment level features.

We propose to measure the semantic similarity between the two kinds of feature to eliminate information redundancy.

For Wikipedia we download the XML corpus, remove xml tags and create a Solr index of all XML articles.
Seed phrases selection (II)

- Let $P$ denotes a segment level feature, $P = \{p_1, p_2, \ldots, p_n\}$.
- We calculate the semantic similarity between $p_i$ and \{p_1, p_2, \ldots, p_n\} as $InfoScore(p_i)$.
- The $p^*$ which has the largest similarity with other features in $P$ will be removed as the redundant feature.
Given two phrases $p_i$ and $p_j$, the variants of three popular co-occurrence measures[6] are defined as below:

\[
WikiDice(p_i, p_j) = \begin{cases} 
0 & \text{if } f(p_i \mid p_j) = 0 \\
& \text{or } f(p_j \mid p_i) = 0 \\
\frac{f(p_i \mid p_j) + f(p_j \mid p_i)}{f(p_i) + f(p_j)} & \text{otherwise}
\end{cases},
\]

where WikiDice is a variant of the Dice coefficient.

\[
WikiJaccard(p_i, p_j) = \frac{\min(f(p_i \mid p_j), f(p_j \mid p_i))}{f(p_i) + f(p_j) - \max(f(p_i \mid p_j), f(p_j \mid p_i))},
\]

where WikiJaccard is a variant of the Jaccard coefficient.
Seed phrases selection (IV)

\[ WikiOverlap(p_i, p_j) = \frac{\min(f(p_i \mid p_j), f(p_j \mid p_i))}{\min(f(p_i), f(p_j))}, \]

where WikiOverlap is a variant of the Overlap(Simpson) coefficient. Linear normalization formula is defined below:

\[ WD_{ij} = \frac{WikiDice_{ij} - \min(WikiDice_k)}{\max(WikiDice_k) - \min(WikiDice_k)}, \]

A linear combination is then used to incorporate the three similarity measures into an overall semantic similarity between two phrases \( p_i \) and \( p_j \), as follows:

\[ WikiSem(p_i, p_j) = (1 - \alpha - \beta)WD_{ij} + \alpha W J_{ij} + \beta W O_{ij}, \]

where \( \alpha \) and \( \beta \) weight the importance of the three similarity measures.
For each segment level feature, we rank the information score defined in Equation 5 for its child node features at phrase level.

\[
InfoScore(p_i) = \sum_{j=1,j\neq i}^{n} WikiSem(p_i, p_j). \tag{6}
\]

Finally, we remove the phrase level feature \( p^* \), which delegates the most information duplicate to the segment level feature \( P \), and it is defined as:

\[
p^* = \arg\max_{p_i \in \{p_1, p_2, \ldots, p_n\}} InfoScore(p_i). \tag{7}
\]
Background Knowledge Bases

- Wikipedia, as background knowledge, has a wider knowledge coverage than WordNet and is regularly updated to reflect recent events.
- On the other hand, as the construction of WordNet follows theoretical model or corpus evidence, it contains rich lexical semantic knowledge.
Feature Generator

**Algorithm 1: GenerateFeatures(S)**

```
input : a set S of seed phrases
output: external features EF

EF ← null
for seed phrase s ∈ S do
    if s.non-stop > 1 then
        if s ∈ Segment level then
            s.Query ← SolrSyntax(s, OR)
        else
            s.Query ← SolrSyntax(s, AND)
        WikiPages ← Retrive(s.Query)
        EF ← EF + Analyze(WikiPages)
    else
        EF ← EF + WordNet.Synsets(s)
return EF
```

**Fig:** *External feature generation scheme*
Feature Selection (I)

Feature filtering for unstructured features:

- Remove features generated from too general *seed phrase* that returns a large number (more than 10,000) of articles from the index corpus.

- Transform features used for Wikipedia management or administration, e.g. “List of hotels” → “hotels”, “List of twins” → “twins”.

- Apply phrase sense stemming using Porter stemmer[24], e.g. “fictional books” → “fiction book”.

- Remove features related to chronology, e.g. “year”, “decade” and “centuries”. 
Feature Selection (II)

To avoid “curse of dimensionality”:

- The number of external features we need to collect is determined by:
  \[ n_2 = \frac{n_1 \times \theta}{1 - \theta}. \]  
  \[(8)\]

- Select one external feature for each seed phrase.
  \[ f_i^* = \arg \max_{f_{ij} \in \{p_{i1}, p_{i2}, \ldots, p_{ik}\}} \text{tf-idf}(f_{ij}). \]  
  \[(9)\]

- The top \( n_2 - m \) features are extracted from the remaining external features based on their frequency.
Data Sets (I)

Reuter-21578:

- We remove the texts which contain more than 50 words and filter those clusters with less than 5 texts or more than 500 texts.
- Thus it leaves 19 clusters comprising 879 texts. The number of texts in each cluster ranges from 6 (the cluster “income”) to 438 (the cluster “acq”).
Data Sets (II)

**Web Dataset** is built to simulate a real web application.

- As the users’ interests are varied, we choose queries of different length according to the statistics of Google Trends during Nov. 26th 2007 – Nov. 25th 2008.

<table>
<thead>
<tr>
<th>query length</th>
<th>One</th>
<th>Two</th>
<th>Three</th>
<th>more</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>4552</td>
<td>19762</td>
<td>6992</td>
<td>5290</td>
</tr>
<tr>
<td>percentage</td>
<td>12.4%</td>
<td>54.0%</td>
<td>19.1%</td>
<td>14.5%</td>
</tr>
</tbody>
</table>

- Ten hot queries are selected.

**Tab:** The selected hot queries in Web Dataset

<table>
<thead>
<tr>
<th>NFL</th>
<th>Amazing Grace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green Bay</td>
<td>Fox News Channel</td>
</tr>
<tr>
<td>60 Minutes</td>
<td>New York Giants</td>
</tr>
<tr>
<td>Total Eclipse</td>
<td>The Dark Knight</td>
</tr>
<tr>
<td>Black Friday</td>
<td>National Economic Council</td>
</tr>
</tbody>
</table>
**K-means and EM** are employed in this study. Six different text representation methods, as defined below:

- **BOW** (baseline 1) : Traditional “bag of words” model with the *tf-idf* weighting schema.
- **BOW+WN** (baseline 2) : *BOW* integrated with additional features from WordNet as presented in [14].
- **BOW+Wiki** (baseline 3) : *BOW* integrated with additional features from Wikipedia as presented in [1].
- **BOW+Know** (baseline 4) : *BOW* integrated with additional features from Wikipedia and WordNet as in baselines 2 and 3.
- **BOF** : The bag of *original features* extracted with the hierarchical view.
- **SemKnow** : Our proposed framework.

We evaluate performance of the methods using *F₁ measure* and *Average Accuracy*. 
**Introduction**

**Proposed Framework**

**Evaluation**

**Conclusion and Future Work**

---

**Performance Evaluation**

**Tab:** Results using *k-means* algorithm

<table>
<thead>
<tr>
<th></th>
<th>Reuters-21578</th>
<th></th>
<th>Web Dataset</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_1$ measure (Impr)</td>
<td>AveAccuracy (Impr)</td>
<td>$F_1$ measure (Impr)</td>
<td>AveAccuracy (Impr)</td>
</tr>
<tr>
<td><strong>BOW</strong></td>
<td>0.471 (N.A.)</td>
<td>0.550 (N.A.)</td>
<td>0.491 (N.A.)</td>
<td>0.563 (N.A.)</td>
</tr>
<tr>
<td><strong>BOW + WN</strong></td>
<td>0.473 (+0.43%)</td>
<td>0.552 (+0.26%)</td>
<td>0.530 (+8.01%)</td>
<td>0.576 (+2.30%)</td>
</tr>
<tr>
<td><strong>BOW + Wiki</strong></td>
<td>0.481 (+2.03%)</td>
<td>0.563 (+2.18%)</td>
<td>0.556 (+13.38%)</td>
<td>0.584 (+3.85%)</td>
</tr>
<tr>
<td><strong>BOW + Know</strong></td>
<td>0.489 (+3.75%)</td>
<td>0.566 (+2.86%)</td>
<td>0.558 (+13.79%)</td>
<td>0.583 (+3.70%)</td>
</tr>
<tr>
<td><strong>BOF</strong></td>
<td>0.473 (+0.33%)</td>
<td>0.551 (+0.19%)</td>
<td>0.520 (+5.95%)</td>
<td>0.570 (+1.24%)</td>
</tr>
<tr>
<td><strong>SemKnow</strong></td>
<td><strong>0.497 (+5.41%)</strong></td>
<td><strong>0.572 (+3.98%)</strong></td>
<td><strong>0.583(+18.81%)</strong></td>
<td><strong>0.586(+4.11%)</strong></td>
</tr>
</tbody>
</table>

**Tab:** Results using *EM* algorithm

<table>
<thead>
<tr>
<th></th>
<th>Reuters-21578</th>
<th></th>
<th>Web Dataset</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_1$ measure (Impr)</td>
<td>AveAccuracy (Impr)</td>
<td>$F_1$ measure (Impr)</td>
<td>AveAccuracy (Impr)</td>
</tr>
<tr>
<td><strong>BOW</strong></td>
<td>0.516 (N.A.)</td>
<td>0.579 (N.A.)</td>
<td>0.521 (N.A.)</td>
<td>0.608 (N.A.)</td>
</tr>
<tr>
<td><strong>BOW + WN</strong></td>
<td>0.525 (+1.72%)</td>
<td>0.585 (+0.99%)</td>
<td>0.540 (+3.59%)</td>
<td>0.626 (+3.02%)</td>
</tr>
<tr>
<td><strong>BOW + Wiki</strong></td>
<td>0.540 (+4.74%)</td>
<td>0.598 (+3.39%)</td>
<td>0.550 (+5.50%)</td>
<td>0.629 (+3.44%)</td>
</tr>
<tr>
<td><strong>BOW + Know</strong></td>
<td>0.542 (+5.13%)</td>
<td>0.607 (+4.54%)</td>
<td>0.556 (+6.74%)</td>
<td>0.635 (+4.41%)</td>
</tr>
<tr>
<td><strong>BOF</strong></td>
<td>0.520 (+0.82%)</td>
<td>0.594 (+2.63%)</td>
<td>0.536 (+2.73%)</td>
<td>0.624 (+2.55%)</td>
</tr>
<tr>
<td><strong>SemKnow</strong></td>
<td><strong>0.548 (+6.28%)</strong></td>
<td><strong>0.622 (+7.51%)</strong></td>
<td><strong>0.569 (+9.07%)</strong></td>
<td><strong>0.670 (+10.20%)</strong></td>
</tr>
</tbody>
</table>
Impact of the parameter $\theta$ on Reuters and Web Dataset using $K$ – $means$ and $EM$ respectively.
**Optimal Results**

**Tab:** Optimal results using two algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Dataset</th>
<th>$F_1$ meas (Impr)</th>
<th>AveAcc (Impr)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>kmeans</strong></td>
<td><strong>Reuters</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BOW</td>
<td>0.471 (N.A.)</td>
<td>0.550 (N.A.)</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td><strong>0.530 (+12.35%)</strong></td>
<td><strong>0.604 (+9.72%)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Webdata</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BOW</td>
<td>0.491 (N.A.)</td>
<td>0.563 (N.A.)</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td><strong>0.640 (+30.39%)</strong></td>
<td><strong>0.607 (+7.83%)</strong></td>
</tr>
<tr>
<td><strong>EM</strong></td>
<td><strong>Reuters</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BOW</td>
<td>0.516 (N.A.)</td>
<td>0.579 (N.A.)</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td><strong>0.578 (+12.02%)</strong></td>
<td><strong>0.672 (+15.40%)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Webdata</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BOW</td>
<td>0.521 (N.A.)</td>
<td>0.608 (N.A.)</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td><strong>0.602 (+16.14%)</strong></td>
<td><strong>0.709 (+16.56%)</strong></td>
</tr>
</tbody>
</table>
In this study, we proposed a novel framework to augment the clustering accuracy of short texts by exploiting the internal and external semantics.

The combination of internal and external semantics well tackled the problems of data sparseness and semantic gap in short texts.

Empirical evaluations demonstrated that our framework significantly outperformed all the baselines including previously proposed knowledge-based short text clustering methods on two datasets.
As this work is for aggregated search, the efficiency of the whole framework should be optimized for real applications.

Moreover, we will explore more tasks in NLP and information retrieval using the internal and external semantics generated by our proposed framework.
Thank you!