A semi-supervised approach to extracting multiword entity names from user reviews

Olga Vechtomova
ovechtom@uwaterloo.ca
Introduction

• Semi-supervised approach to extracting entities (both single words and multiword units) of a specific semantic class from user-written reviews
• Can be applied to extract different classes of entities
• Task: extraction of dish names from restaurant reviews
• Identification and removal of subjective modifiers
• Novel use of BM25 as distributional similarity measure
• Comparison with other similarity measures
Computing similarity between seeds and single words

- Pre-processing: perform dependency parsing of the corpus (Stanford parser)
- Examples of dependency triples:
  - `gnocchi with brown butter and crispy sage leaves`
    - amod NN JJ (butter, brown)
    - prep_with NN NN (gnocchi, butter)
    - nn NNS NN (leaves, crispy)
    - nn NNS NN (leaves, sage)
    - prep_with NN NNS (gnocchi, leaves)
    - conj_and NN NNS (butter, leaves)
Computing similarity between seeds and single words

- Build feature vectors
  - Take all dependency triples containing seed words
  - Transform them into vector features

Seed word: butter
  - amod NN JJ (butter, brown)
  - prep_with NN NN (gnocchi, butter)
  - conj_and NN NNS (butter, leaves)
  - amod NN JJ (X, brown)
  - prep_with NN NN (gnocchi, X)
  - conj_and NN NNS (X, leaves)

- A vector for each word consists of these features and their frequencies of occurrence with this word
BM25-based distributional similarity measure

- Compute similarity between the vectors of each candidate and seed using BM25 with query weights (Spärck Jones et al., 2000):

\[ QACW_{c,s} = \sum_{f=1}^{F} \frac{TF(k_1 + 1)}{K + TF} \times QTF \times IDF_f \]

- \( F \) - number of features in common between candidate word \( c \) and seed \( s \)
- \( TF \) – frequency of occurrence of feature \( f \) with the candidate word
- \( QTF \) – frequency of occurrence of \( f \) with the seed
BM25-based distributional similarity measure

\[ IDF_f = \log \frac{N}{n_f} \]

- \( n_f \) - number of candidate word vectors the feature \( f \) occurs in
- \( N \) - number of candidate word vectors
Other distributional similarity methods

• Lin’s measure (Lin 1998)
  – uses PMI to calculate association between a word and a feature (dependency triple)
• Weeds and Weir (2003)
  – adapt the concepts of precision and recall to compute similarity
• Kotlerman et al. (2009)
  – propose directional (asymmetric) measure (balAPinc), aimed at finding more specialized terms
Computing similarity between seeds and single words

- Feature-Seed co-occurrence threshold ($t$):
  - Use only those features that occur with at least $t$ seeds
Effect of seed threshold \((t)\) on MAP (Nouns)
Effect of seed threshold \((t)\) on MAP (Adjectives)

MAP vs. seed threshold \((t)\):
- lin
- weeds
- balapinc
- bm25

University of Waterloo
Removing subjective modifiers

- delicious italian pizza

- Use top ranked adjectives as subjective (parameter $a$)
- In each NP find the rightmost occurrence of a subj. adjective
- Remove all words preceding and including this adjective
- Rationale: modifiers in English are used in a certain order
  - opinion, size, age, shape, colour, origin, material, purpose
Effect of the number of top ranked adjectives removed on performance

![Graph showing the effect of the number of top ranked adjectives removed on performance. The x-axis represents the number of top ranked adjectives (a), ranging from 0 to 400, with the label "all" at the end. The y-axis represents MAP (mean average precision), ranging from 0 to 0.45. Different lines represent various models: lin, weeds, balapinc, and bm25. The graph indicates that as the number of top ranked adjectives removed increases, the MAP decreases.](image-url)
Ranking noun phrases

- Intuition: the further away the noun is from the NP head the less its score should contribute to the NP score
  - “restaurant pizza” and “pizza restaurant”
- Discount noun scores based on distance from the end of NP

<table>
<thead>
<tr>
<th>Discount Type</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log-linear</td>
<td>( D = 1 - \log_{10}(d_i) )</td>
</tr>
<tr>
<td>Linear</td>
<td>( D = 1 - ((d_i - 1) \times 0.1) )</td>
</tr>
<tr>
<td>No discount</td>
<td>( D = 1 )</td>
</tr>
<tr>
<td>0.5 discount</td>
<td>( D = \begin{cases} 1 &amp; \text{if } d_i = 1 \ 0.5 &amp; \text{otherwise} \end{cases} )</td>
</tr>
</tbody>
</table>
Ranking noun phrases

\[ NPscore = \frac{\sum_{i=1}^{n} DW_i}{n} \]

- \( w_i \) - seed-similarity score of the noun
- \( D \) - discount function
- \( n \) - number of words in the NP
Effect of the discount factors on performance

MAP

lin weeds balapinc bm25

0.3 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39

- 0.5 discount
- linear
- log
- no discount
Evaluation

• Corpus of 157,865 restaurant reviews from Citygrid
• 2 annotators labeled dish names and subjective adjectives in 600 reviews:
  – 1000 unique dish names (MWUs and single nouns)
  – 573 unique single word food/dish names
  – 472 unique subjective adjectives
• Seed sets:
  – 20 seed sets, 10 single words each
### Results

- Ranking of single noun food/dish names

<table>
<thead>
<tr>
<th>Run</th>
<th>MAP</th>
<th>P@50</th>
<th>P@100</th>
<th>P@200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin (t=3)</td>
<td>0.5255</td>
<td>0.968</td>
<td>0.893</td>
<td>0.7755</td>
</tr>
<tr>
<td>Weeds (t=1)</td>
<td>0.5501</td>
<td>0.886</td>
<td>0.795</td>
<td>0.7445</td>
</tr>
<tr>
<td>balAPinc (t=2)</td>
<td>0.5836</td>
<td>0.964</td>
<td><strong>0.91</strong></td>
<td><strong>0.82</strong></td>
</tr>
<tr>
<td>BM25 (t=1)</td>
<td>0.5705</td>
<td><strong>0.97</strong></td>
<td>0.893</td>
<td>0.811</td>
</tr>
</tbody>
</table>
Results

- Ranking of adjectives

<table>
<thead>
<tr>
<th>Run</th>
<th>MAP</th>
<th>P@50</th>
<th>P@100</th>
<th>P@200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin (t=3)</td>
<td>0.7442</td>
<td>0.914</td>
<td>0.883</td>
<td>0.842</td>
</tr>
<tr>
<td>Weeds (t=1)</td>
<td>0.7334</td>
<td>0.916</td>
<td>0.878</td>
<td>0.835</td>
</tr>
<tr>
<td>balAPinc (t=2)</td>
<td>0.7296</td>
<td>0.892</td>
<td>0.859</td>
<td>0.812</td>
</tr>
<tr>
<td>BM25 (t=1)</td>
<td><strong>0.7744</strong></td>
<td><strong>0.922</strong></td>
<td><strong>0.889</strong></td>
<td><strong>0.861</strong></td>
</tr>
</tbody>
</table>
Results

- Ranking of multiword dish names

<table>
<thead>
<tr>
<th>Run</th>
<th>MAP</th>
<th>P@50</th>
<th>P@100</th>
<th>P@200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin (a=50)</td>
<td>0.3738</td>
<td>0.92</td>
<td>0.831</td>
<td>0.759</td>
</tr>
<tr>
<td>Weeds (a=50)</td>
<td>0.3483</td>
<td>0.854</td>
<td>0.787</td>
<td>0.684</td>
</tr>
<tr>
<td>balAPinc (a=50)</td>
<td>0.3742</td>
<td>0.886</td>
<td>0.814</td>
<td>0.7245</td>
</tr>
<tr>
<td>BM25 (a=100)</td>
<td><strong>0.3814</strong></td>
<td>0.832</td>
<td>0.779</td>
<td>0.715</td>
</tr>
</tbody>
</table>
Effect of the number of seeds on performance

- bm25
- balapinc

MAP

- 5-seeds
- 10-seeds
- 15-seeds
- 20-seeds
Future work

• Better method to detect boundaries of multiword entity names is needed, e.g.
  – “arugula salad with fresh parmesan”
  – “made by hand fries in a sundae dish with three different dips”
  – “pasta with lamb, olives, goat cheese and rosemary”

• Application to other types of entities
  – Could be applied to identify different types of entities in user reviews
  – Promising results of a small-scale evaluation of other aspects of restaurant reviews (e.g., ambiance/atmosphere; people/staff)
Questions?