Distributed Representations of Geographically Situated Language

David Bamman Chris Dyer Noah A. Smith

Presenter: Konstantinos Pappas
“Who?”

David Bamman
Assistant Professor

Chris Dyer
Assistant Professor

Noah Smith
Associate Professor

[Logos of the universities: University of California, Berkeley, Carnegie Mellon University, University of Washington]
1. Incorporate contextual information (geography) in learning vector-space representations of situated language.
- contextual variable: 51 US states
- dataset: 93M tweets - 1B words
- representation: embeddings (Mikolov et al. 2013)
- evaluation: qualitatively (manual inspection) and quantitatively (semantic similarity)
“WHY?” #1
“WHY?” #2

In particular, how is a word’s meaning shaped by its geography?
Distributional hypothesis

Words that occur in similar contexts tend to have similar meanings (Harris, 1954; Firth, 1957; Deerwester et al., 1990). If words have similar row vectors in a word-context matrix, then they tend to have similar meanings.

1-modal learning:
- textual context
You shall know a word by the company it keeps . . .
Firth (1957)

Multimodal learning:
- object info e.g. visual cues
- speaker info e.g. location
An extension of the “skip-gram” model (Mikolov et al. 2013)

Defines a set of contextual variables:
Cstate = (AK, AL, ..., WY)
Model details

One global embedding matrix $W$.

Another 51 matrices which capture the effect that each variable value has on each word in the vocabulary.

Each deviation indicates how that common representation should shift in the $k$-dimensional space when used in each state.

Backpropagation, L2 regularization.
IMPLEMENTATION

github: https://github.com/dbamman/geoSGLM

GeoSGLM: Code for learning geographically-informed word embeddings.

To run, adjust the input/output parameters in run.sh and execute it.
<table>
<thead>
<tr>
<th>id</th>
<th>location</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td>480326347508969000</td>
<td>PA</td>
<td>There is a great research question in how long a sequence of blog comments can go before it descends into madness <a href="http://t.co/NFqKgaZRuO">http://t.co/NFqKgaZRuO</a></td>
</tr>
<tr>
<td>472023364908118000</td>
<td>PA</td>
<td>So much easier than hunting through individual websites: using Google Scholar to get BibTeX citations <a href="http://t.co/H2inkMGMom">http://t.co/H2inkMGMom</a></td>
</tr>
<tr>
<td>105039889808109000</td>
<td>PA</td>
<td>Just discovered Conflict Kitchen in Pittsburgh - brilliant idea that needs to catch on in other cities. <a href="http://t.co/FkSLGD9">http://t.co/FkSLGD9</a></td>
</tr>
</tbody>
</table>
The vocab file contains the maximal set of words to learn representations for.

If a word is not in this list, then don't learn a representation for it.

This list is further filtered in the code to only include words that are seen at least 5 times in the data, and a maximum of the $\text{MAXVOCAB}$ most frequent terms.
ETC

FEATUREFILE=data/states.txt

OUTFILE=data/out.embeddings

DIMENSIONALITY=100
Dimensionality specifies the size of the learned word representations.

L2=0.0001
L2 regularization parameter.
Similarity

For a given query q, you can view the terms most similar to q in all 51 states using scripts/findNearest.py

python scripts/findNearest.py $OUTFILE
## Qualitative Analysis #1

<table>
<thead>
<tr>
<th>Kansas</th>
<th>Massachusetts</th>
</tr>
</thead>
<tbody>
<tr>
<td>term</td>
<td>cosine</td>
</tr>
<tr>
<td>wicked</td>
<td>1.000</td>
</tr>
<tr>
<td>evil</td>
<td>0.884</td>
</tr>
<tr>
<td>pure</td>
<td>0.841</td>
</tr>
<tr>
<td>gods</td>
<td>0.841</td>
</tr>
<tr>
<td>mystery</td>
<td>0.830</td>
</tr>
<tr>
<td>spirit</td>
<td>0.830</td>
</tr>
<tr>
<td>king</td>
<td>0.828</td>
</tr>
<tr>
<td>above</td>
<td>0.825</td>
</tr>
<tr>
<td>righteous</td>
<td>0.823</td>
</tr>
<tr>
<td>magic</td>
<td>0.822</td>
</tr>
</tbody>
</table>

Table 1: Terms with the highest cosine similarity to *wicked* in Kansas and Massachusetts.
# Qualitative Analysis #2

<table>
<thead>
<tr>
<th></th>
<th>California</th>
<th>New York</th>
</tr>
</thead>
<tbody>
<tr>
<td>term</td>
<td>cosine</td>
<td>term</td>
</tr>
<tr>
<td>city</td>
<td>1.000</td>
<td>city</td>
</tr>
<tr>
<td>valley</td>
<td>0.880</td>
<td>suburbs</td>
</tr>
<tr>
<td>bay</td>
<td>0.874</td>
<td>town</td>
</tr>
<tr>
<td>downtown</td>
<td>0.873</td>
<td>hamptons</td>
</tr>
<tr>
<td>chinatown</td>
<td>0.854</td>
<td>big city</td>
</tr>
<tr>
<td>south bay</td>
<td>0.854</td>
<td>borough</td>
</tr>
<tr>
<td>area</td>
<td>0.851</td>
<td>neighborhood</td>
</tr>
<tr>
<td>east bay</td>
<td>0.845</td>
<td>downtown</td>
</tr>
<tr>
<td>neighborhood</td>
<td>0.843</td>
<td>upstate</td>
</tr>
<tr>
<td>peninsula</td>
<td>0.840</td>
<td>big apple</td>
</tr>
</tbody>
</table>

Table 2: Terms with the highest cosine similarity to *city* in California and New York.
Quantitative evaluation - Set up

7 categories

1. city - most populous city/state
2. state - state name
3. football - NFL team names
4. basketball - NBA team names
5. baseball - MLB team names
6. hockey - NHL team names
7. park - US national parks

3 models

1. Joint: global representation for each word + a deviation per state
2. Individual: each state one model
3. -GEO: one model from the whole US
Average cosine similarity for all models across all categories, with 95% confidence intervals on the mean.
The paper provides an extension to vector-space representations that can take into account the context in which it is uttered.

Implements three models: joint, individual, normal.

Provides two different kinds of evaluation of the models.

Discusses possible extensions and applications of this tool.
NOTES

- ACL 2014 (+)
- Mentions that this tool for revealing periodic and historical influences on lexical semantics, but provides no evidence (-)
- Provides online implementation of the system (+)
Questions

Can we realistically find enough data for each contour that we are interested? E.g. a particular year?

How can these new embeddings be used for IR?

Would it make sense to create different embeddings per gender? Per age of author?