Discriminative keyword extraction

Márius Šajgalík
marius.sajgalik@stuba.sk
FIIT STUBA
Keywords

Metadata

**Concise** representation of text, images, etc.

Help to organize data
Discriminative keywords

Keywords can be used to discriminate between categories

Higher information value

Avoid generally important words
Our method of discriminative keyword extraction

1. Candidate phrase extraction
2. Computation of distributed representation for candidate phrases
3. Substitution of phrases by most similar words
4. Computation of discriminative metric
5. Keyword selection (ranking)
6. Computation of document vector
Distributed representation of words

Words mapped to vectors
  - word vectors, word embeddings

Distributed word features

Distributional hypothesis

“Words are similar if they appear in similar context.”
Motivation: Why distributed representation?

audio  images  text

DENSE  DENSE  SPARSE
Properties of Distributed Representation

Multiple degrees of similarity

We can do **vector operations**

\[
\text{vector("snow") + vector("ball")} \approx \text{vector("snow ball")}
\]

**Syntactic relations**

\[
\text{vector("biggest")} - \text{vector("big")} + \text{vector("small")} \approx \text{vector("smallest")}
\]

**Semantic relations**

\[
\text{vector("Paris")} - \text{vector("France")} + \text{vector("Germany")} \approx \text{vector("Berlin")}
\]
Table 1: Examples of five types of semantic and nine types of syntactic questions in the Semantic-Syntactic Word Relationship test set.

<table>
<thead>
<tr>
<th>Type of relationship</th>
<th>Word Pair 1</th>
<th>Word Pair 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common capital city</td>
<td>Athens</td>
<td>Greece</td>
</tr>
<tr>
<td>All capital cities</td>
<td>Astana</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>Currency</td>
<td>Angola</td>
<td>kwanza</td>
</tr>
<tr>
<td>City-in-state</td>
<td>Chicago</td>
<td>Illinois</td>
</tr>
<tr>
<td>Man-Woman</td>
<td>brother</td>
<td>sister</td>
</tr>
<tr>
<td>Adjective to adverb</td>
<td>apparent</td>
<td>apparently</td>
</tr>
<tr>
<td>Opposite</td>
<td>possibly</td>
<td>impossibly</td>
</tr>
<tr>
<td>Comparative</td>
<td>great</td>
<td>greater</td>
</tr>
<tr>
<td>Superlative</td>
<td>easy</td>
<td>easiest</td>
</tr>
<tr>
<td>Present Participle</td>
<td>think</td>
<td>thinking</td>
</tr>
<tr>
<td>Nationality adjective</td>
<td>Switzerland</td>
<td>Swiss</td>
</tr>
<tr>
<td>Past tense</td>
<td>walking</td>
<td>walked</td>
</tr>
<tr>
<td>Plural nouns</td>
<td>mouse</td>
<td>mice</td>
</tr>
<tr>
<td>Plural verbs</td>
<td>work</td>
<td>works</td>
</tr>
</tbody>
</table>

word2vec

Open sourced tool from Google researchers

Learns **word embeddings** from **raw text**

Efficient parallel implementation in C with pretrained English model

https://code.google.com/archive/p/word2vec/

**Gensim** - Python implementation for multiple tasks

http://radimrehurek.com/gensim/models/word2vec.html
Visualization of word embeddings with t-SNE
Distributed representation of ???

- Web pages visited by users
- Documents bookmarked by users
- Products viewed/bought by customers
Our method of discriminative keyword extraction

1. Candidate phrase extraction
2. Computation of **distributed representation** for candidate phrases
3. Substitution of phrases by most similar words
4. Computation of **discriminative metric**
5. Keyword selection (ranking)
6. Computation of document vector
Discriminative metrics

Based on categorisation of text documents

Most of the metrics can be expressed by ABCD statistics

<table>
<thead>
<tr>
<th>Frequency</th>
<th>word W</th>
<th>other words</th>
</tr>
</thead>
<tbody>
<tr>
<td>in category CAT</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>in other categories</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>
### Discriminative metrics (2)

<table>
<thead>
<tr>
<th>Metric name</th>
<th>Function expressed in terms of ABCD statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>rf</td>
<td>( \log(2 + \frac{A}{\max(C,1)}) )</td>
</tr>
<tr>
<td>tds</td>
<td>( \frac{A/(A+B)}{(A+C)/N} )</td>
</tr>
<tr>
<td>ig</td>
<td>( \frac{\frac{A}{N} \times \frac{\log(A \times N)}{(A+C) \times (A+B)}}{\frac{B}{N} \times \frac{\log(B \times N)}{(B+D) \times (A+B)} \times \frac{C}{N} \times \frac{\log(C \times N)}{(A+D) \times (C+D)} \times \frac{D}{N} \times \frac{\log(D \times N)}{(B+D) \times (C+D)} } )</td>
</tr>
<tr>
<td>gr</td>
<td>( \frac{-ig}{(\frac{A+B}{N} \times \log(\frac{A+B}{N}) + \frac{C+D}{N} \times \log(\frac{C+D}{N})} )</td>
</tr>
<tr>
<td>( \chi^2 )</td>
<td>( N \times \frac{(A \times D - B \times C)^2}{(A+D)(B+C)(A+B)(C+D)} )</td>
</tr>
<tr>
<td>idf</td>
<td>( \log(\frac{N}{A+C}) )</td>
</tr>
</tbody>
</table>
Our method of discriminative keyword extraction

1. Candidate phrase extraction (noun phrases)
2. Computation of **distributed representation** for candidate phrases
3. Substitution of phrases by most similar words (kNN in vector space)
4. Computation of **discriminative metric**
5. Keyword selection (just sorting)
6. Computation of document vector (a sum)
Computation of distributed representation for candidate phrases

Summation of vectors of individual terms

Terms - words and short phrases - entries in the dictionary

Dynamic programming to compute the vector of ambiguous phrases
Our method of discriminative keyword extraction (2)

Article about a protest in harbours
● category ships in Reuters-21578 dataset

pirates, pirate, steamer, anchorage, Ship, sail, ferry, destroyer, Ships, destroyers
Our method of discriminative keyword extraction (3)

Manual evaluation
● Part of 20newsgroups dataset
● TF-RF baseline

Automatic evaluation
● 4 different datasets
● Analysis of influence of the individual parameters
  ○ Choice of discriminative metric
  ○ Number of extracted words
Our method of discriminative keyword extraction (4)
Our method of discriminative keyword extraction (5)

+ Capturing **abstract** concepts
+ **Discriminative** representation
+ **Justification** by distributed representation
Our method of discriminative keyword extraction (5)

+ Capturing abstract concepts
+ Discriminative representation
+ Justification by distributed representation

+ Weakly language dependent
+ Scalability
+ Multiple concepts mixing together
  - Multiple concepts mixing together
Modelling user interests

User model as a tripartite graph (Mika, 2007)
\[ G = (V, E) \]
\[ V = U \cup W \cup D \]
\[ E = \{(u, w, d) | u \in U, w \in W, d \in D\} \]

Users as categories

Using our method of discriminative keywords extraction
Modelling user interests (2)

Discovering value from community activity on focused question answering sites: a case study of stack overflow

Hybrid Web Recommender Systems

Context-aware query classification
Modelling user interests (3)

Evaluated on 2 datasets

**Annota**
- Bookmarked research articles in digital library

**Brumo**
- Web browsing logs
Modelling user interests (4)

- (Dis)advantages of the method of discriminative keyword extraction
  + Personalised keywords
  + Fast automatic evaluation of classification
Diversion: Neural networks architectures
Neural networks as a black box

Traditional architectures of the last century

Mostly supervised learning
Neural networks as a white box

Unconventional architectures

Multiple outputs

Output in hidden layers

Network modularisation
“word2vec” architectures - output in hidden layer
Going deeper with convolutions

Modularised

Multiple outputs - controlling hidden layers
Inception module - naive
Inception module - with dimensionality reductions
Rethinking the Inception Architecture


Factorization into smaller and asymmetric convolutions

Grid size reductions

Label-smoothing regularization
Rethinking the Inception Architecture (2)
PolyNet: A Pursuit of Structural Diversity in Very Deep Networks


PolyInception modules

Initialization

Residual scaling

Stochastic paths
PolyNet: A Pursuit of Structural Diversity in Very Deep Networks (2)

(a) poly-2  (b) poly-2  (c) mpoly-2  (d) 2-way
ResNet - a universal module

\[
F(x) + x
\]

\[
\text{relu}
\]

\[
\text{identity}
\]

\[
\text{weight layer}
\]

\[
\text{weight layer}
\]

\[
\text{relu}
\]

\[
F(x)
\]
Unsupervised keyword extraction
Unsupervised keyword extraction

The real output in hidden layer

Two objectives

● We want feature vectors representing keywords
● We want discriminative feature vectors
Unsupervised keyword extraction

words in document

word2vec

CNN/LSTM

keywords

cosine distance

min

categorisation (softmax)
Unsupervised keyword extraction

Works for short texts
- Beware of duplicates

Standard categorisation datasets are challenging

We tried pointer networks

Investigating memory networks
Pointer networks

Pointer sentinel mixture model

Memory networks

http://www.thespermwhale.com/jaseweston/icml2016/
Summary

Focus on discriminative features

Distributed representation

Neural networks + unsupervised learning

Neural network architectures as a white box