Feature Associations in Graph Structures for Unsupervised Entity Disambiguation

Roman Kern
rkern@know-center.at

WIQE10 / 2010-09-14
Overview

Motivation

Approach
  Model
  Algorithm

Applications
  Tag Recommender
  Machine Translation
  Information Retrieval
  Crosslingual Plagiarism Detection
  Unsupervised Entity Disambiguation

Conclusions
Anatomy of a Knowledge Discovery Application

- Input: Data stored in repositories
  - Structured vs. unstructured data
  - Textual vs. multi-media content
  - Single vs. multiple repositories
- Preprocessing of input into data-structures suitable for algorithms
- Apply algorithms on data-structures
- Output: Visualize & store result

[Fayyad et al. 96]
Feature Engineering

- Transform data into features
  - Feature extraction: Words, Syntax, Statistics, ...
  - Feature representation: Plain Text, Arrays, Matrix, Graph, ...
- Specific algorithms need specific data-structures
  - High Level: Information Extraction, Classification, Clustering, Information Retrieval, ...
  - Low Level: SVD, EVD, SVM, HAC, BM25, LSA, LDA, TFIDF, CRF, KNN, HMM, ...

- Example: Vector Space Model
  - Input: Documents
  - Features: Terms
  - Data-structure: Matrix
Feature Associations

Relationships between features

- Additional transformation step
- Network of features
- *Example*: Term co-occurrences

Goal: Framework for feature associations

- Calculate feature associations
- Provide data-structure for feature associations
- Support feature engineering
  - Feature analysis
  - Feature synthesis
- Support application development
  - Common data-structure for algorithms from various domains
Algorithmic Issues

How to represent different features?

- Associations between features of different types
- More features could lead to better results
  - Example: String kernels for classification
- But: More features definitely lead to more expensive computation

How to integrate external knowledge?

- WordNet, ConceptNet, Linked Data, LDAP, ...

How to calculate the association weight?

- Correlation, statistical tests, probabilities, ...
Practical Issues

How to deal with data that does not fit into main memory?

- Enterprise scale
- *Example*: English Wikipedia: $10^7$ documents, $10^6$ terms

How to integrate (exploit) heuristics?

- Strong (naive) independence assumption, Zipf’s law, Heaps’ law, small world networks, distributional hypothesis, ...
Approach - Overview

Feature association framework

- Calculate feature associations
  - Input: Extracted features in graph-like structures
  - Output: Feature network
- Access feature associations
  - Traverse feature network

Solves algorithmic and practical issues

- Provides an scalable algorithmic approach for large scale datasets
- Flexible to allow the integration of rich set of features and external sources
- Allows the integration of a range of graph operations to build the association network
Approach - Generalizations

Starting Point

- Vector Space Model: Inverted Index
- Simple feature representation: $Matrix_{Documents \times Terms}$
- Simple feature operations: $cos_{\text{sim}}(row(M, i), row(M, j))$

Generalize Feature Representation

- Generalization of the simple matrix model
- Allows integration of additional information, e.g. term positions, external sources, linguistic annotations, ...

Generalize Feature Operations

- Generalization of the operations on the features
- Allows integration of algorithms, e.g. Levenshtein edit distance, SVD, clustering, ...
Generalize Feature Representation

Feature Data-Structure

► Matrix can be transformed into a bi-partite graph

\[
\begin{pmatrix}
1 & 1 \\
1 & \\
1 & \\
1 & 1 \\
1 & 1 \\
\end{pmatrix}
\]

► Bi-partite graph can be generalized to a n-partite graph

 Documents

 Terms

 Metadata

- Documents
- Terms
- Metadata
Generalize Feature Operations

Matrix Feature Function

- Matrix multiplication

\[ M_{n,n} = M_{m,n} \times M_{m,n}^T \]

- General matrix transformation

\[ M_{n',m'} = f(M_{m,n}, M_{m,n}^T) \]

- Simple Example - Matrix transposition

\[ M_{n,m} = M_{m,n}^T \]

Not only for matrices, but graphs too.
Feature Operation Functions

- Feature association function - $f(i, j)$

$$f(node_i, node_j) = w_{global}(a(node_i, node_j), G)$$

$$a(node_i, node_j) = w_{aggregate}(\{w_{combine}(l(i), l(j))\})$$

$$l(node_x) = w_{local}(node_x, L)$$

- Input variables
  - Local - $\mathcal{L}$: word-forms, position, term frequency, document length, ...
  - Global - $\mathcal{G}$: document frequency, dispersion, co-occurrence count, average document length, ...

- Examples
  - Cosine Similarity, Jaccard, Windowed Co-Occurrence, Poisson, Pascal, Binomial, PMI, Conditional Probability, Conditional Entropy, Mutual Information, $\chi^2$, Log Odds, ...
Properties

Runtime Complexity

- Runtime complexity of $\mathcal{O}(n^2 \times m)$
- Wikipedia: $10^{19}$ Operations

Algorithm

- Number of heuristics to keep computation feasible
  - Expects power law
  - Expects globally sparse, but locally highly connected
- Execution can be done in parallel
- Map-Reduce friendly
Implementation tailored towards contemporary computer architecture: Memory Access $\ll$ Disk Access

- CPU & IO bound
Item based tag recommender system
- Input data: Folksonomy (Flickr subset)
- Features: Tags, Photos
- Data-Structure: Bipartite Graph (Matrix)
- Feature function: \( w_{i,j} = \frac{\text{sharedPhotos}_{i,j}}{\text{mean} (\text{photoCount}_i, \text{photoCount}_j)} \)
- Feature association retrieval: Lookup

Recommending tags for pictures based on text, visual content and user context. Lindstaedt, Pammer, Moerzinger, Kern, Mülner, and Wagner [2008]
Folksonomy Analysis

- Statistical Analysis of a Folksonomy
  - Input data: Folksonomy (Flickr subset) stored in SQL-database
  - Features: Tags, Photos, Users, Title, Description and Comments
  - Data-Structure: N-Bipartite Graph
  - Feature function: Cosine
  - Feature association retrieval: Spreading Activation, Distance

Extending Folksonomies for Image Tagging.
Kern, Granitzer, and Pammer [2008]
Machine Translation

- Word alignment for query translation
- Input data: multilingual corpus (Wikipedia, Europarl)
- Features: English words, Spanish words
- Data-Structure: article aligned corpus, 3-partite graph
- Feature function: Cosine, Correlation, TFIDF
- Feature association retrieval: Spreading Activation

Crosslanguage Retrieval based on Wikipedia Statistics.
Juffinger, Kern, and Granitzer [2008a]
Exploiting Cooccurrence on Corpus and Document Level for Fair Crosslanguage Retrieval. Juffinger, Kern, and Granitzer [2008b]
Global query expansion for cross-lingual information retrieval

- Textual corpus (Glasgow Harald, LA Times)
- English words & positions, PMI
- Monolingual Performance

<table>
<thead>
<tr>
<th>Query Expansion</th>
<th>MAP</th>
<th>GMAP</th>
<th>Wilcoxon</th>
<th>Randomized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.4022</td>
<td>0.1805</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WSD WordNet</td>
<td>0.4070</td>
<td>0.1869</td>
<td>0.0119</td>
<td>0.0147</td>
</tr>
<tr>
<td>Co-occurrence Terms</td>
<td>0.4170</td>
<td>0.1864</td>
<td>0.0001</td>
<td>0.0196</td>
</tr>
</tbody>
</table>

Crosslingual Performance

<table>
<thead>
<tr>
<th>Query Expansion</th>
<th>MAP</th>
<th>GMAP</th>
<th>Wilcoxon</th>
<th>Randomized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.2885</td>
<td>0.0762</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WSD WordNet</td>
<td>0.2933</td>
<td>0.0773</td>
<td>0.2187</td>
<td>0.0056</td>
</tr>
<tr>
<td>Co-occurrence Terms</td>
<td>0.2917</td>
<td>0.0718</td>
<td>0.0090</td>
<td>0.0252</td>
</tr>
</tbody>
</table>

Application of Axiomatic Approaches to Crosslanguage Retrieval.
Kern, Juffinger, and Granitzer [2009a]
Evaluation of Axiomatic Approaches to Crosslanguage Retrieval.
Kern, Juffinger, and Granitzer [2009b]
Crosslingual Plagiarism Detection

- Goal: Retrieve word translation candidates to detect crosslingual plagiarism
- Features: word alignment candidates
- Data-Structure: sentence aligned corpus (Europarl)
- Feature Function: HMM based word alignment algorithm (BerkeleyAligner)

*External and Intrinsic Plagiarism Detection using a Cross-Lingual Retrieval and Segmentation System*
Muhr, Kern, Zechner, and Granitzer [2010]*
Word sense induction and discrimination

- **Goal:** Identify the individual senses of an ambiguous word and label unseen instance with one of them
- **Features:** Grammatical dependencies, (expanded) sentence phrase terms
Word sense induction and discrimination

▶ Sense induction:
  ▶ Extract local sub-graphs
  ▶ Cluster sub-graphs
  ▶ Generate new features out of existing features
  ▶ Senses are additional features in the feature association network

▶ Sense discrimination:
  ▶ Distance based similarity search

# Results

## Classification

<table>
<thead>
<tr>
<th>Domain</th>
<th>Application</th>
<th>Key Results</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Web</td>
<td>Tag Recommender</td>
<td>Proof of concept</td>
<td>Easy exchange of similarity function</td>
</tr>
<tr>
<td>Social Web</td>
<td>Folksonomy Analysis</td>
<td>Deeper understanding of folksonomies, Base for recommender systems</td>
<td>Integration of additional features</td>
</tr>
<tr>
<td>Information Retrieval</td>
<td>Query Translation</td>
<td>Good translation performance</td>
<td>Simple mapping of aligned documents, Efficient lookup</td>
</tr>
<tr>
<td>Information Retrieval</td>
<td>Query Expansion</td>
<td>Improved Performance over baseline</td>
<td>Integration of task-specific weighting function</td>
</tr>
<tr>
<td>Natural Language Processing</td>
<td>Crosslingual Plagiarism Detection</td>
<td>Lookup runtime performance</td>
<td>Integration into real-world systems</td>
</tr>
<tr>
<td>Natural Language Processing</td>
<td>Word Sense Induction and Discrimination</td>
<td>State-of-the-art performance</td>
<td>Integration of different features, Integration of different algorithms</td>
</tr>
</tbody>
</table>
Conclusions

- Algorithmic approach
  - Calculate feature associations
  - Traverse feature association networks
- Goals
  - Scalable
  - Flexible
  - Usable
- Applications
  - Different domains related to knowledge discovery
  - Real-world benefit
The End

Thank you!
References


