Continuous Vector Spaces for Cross-Language NLP Applications

Rafael E. Banchs
Human Language Technology Department,
Institute for Infocomm Research, Singapore

November 1, 2016
Austin, Texas, USA.
Tutorial Outline

PART I

• Basic Concepts and Theoretical Framework (≈45 mins)
• Vector Spaces in Monolingual NLP (≈45 mins)

PART II

• Vector Spaces in Cross-language NLP (≈70 mins)
• Future Research and Applications (≈20 mins)
Motivation

• The mathematical metaphor offered by the geometric concept of distance in vector spaces with respect to semantics and meaning has been proven to be useful in monolingual NLP applications.

• There is some recent evidence that this paradigm can also be useful for cross-language NLP applications.
Objectives

The main objectives of this tutorial are as follows:

• To introduce the basic concepts related to distributional and cognitive semantics

• To review some classical examples on the use of vector space models in monolingual NLP applications

• To present some novel examples on the use of vector space models in cross-language NLP applications
Section 1

Basic Concepts and Theoretical Framework

- The Distributional Hypothesis
- Vector Space Models and the Term-Document Matrix
- Association Scores and Similarity Metrics
- The Curse of Dimensionality and Dimensionality Reduction
- Semantic Cognition, Conceptualization and Abstraction
Distributional Hypothesis

“a word is characterized for the company it keeps” *
(meaning is mainly determined by the context rather than from individual language units)

Distributional Structure

Meaning as a result of language’s Distributional Structure ... or vice versa?

“... if we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C.” *

“In the language itself, there are only differences” **


** Saussure, F. (1916) Course in General Linguistics
Not everyone is happy... 😞

**Argument against...**

- Meaning involves more than language:
  - Images and experiences that are beyond language
  - Objects, ideas and concepts in the minds of the speaker and the listener

**Counterargument...**

- “if extralinguistic factors do influence linguistic events, there will always be a distributional correlate to the event that will suffice as explanatory principle” *

---

Not everyone is happy... 😞

Argument against...

- The concept of semantic difference (or similarity) is too broad to be useful !!!

Counterargument ...

- Semantic relations “are not axiomatic, and the broad notion of semantic similarity seems perfectly plausible” *

Functional Differences

• Functional differences across words are fundamental for defining the notion of meaning

• Two different types of functional differences between words can be distinguished: *
  ▪ Syntagmatic relations: Explain how words are combined (co-occurrences)
  ▪ Paradigmatic relations: Explain how words exclude each other (substitutions)

* Saussure, F. (1916) Course in General Linguistics
Orthogonal Dimensions

Paradigmatic

some scientists look smart
few people feel dumb
most citizens seem gifted
many lawyers are savvy
The Term-context Matrix

D1: dogs are animals
D2: cats are animals
D3: orchids are plants
D4: roses are plants

<table>
<thead>
<tr>
<th></th>
<th>Animals</th>
<th>Are</th>
<th>Cats</th>
<th>Dogs</th>
<th>Orchids</th>
<th>Plants</th>
<th>Roses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cats</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dogs</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orchids</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plants</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roses</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
## Paradigmatic Relation Matrix

### Top Paradigmatic Pairs
- (dogs, cats)
- (orchids, roses)

<table>
<thead>
<tr>
<th></th>
<th>Animals</th>
<th>Are</th>
<th>Cats</th>
<th>Dogs</th>
<th>Orchids</th>
<th>Plants</th>
<th>Roses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cats</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dogs</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orchids</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plants</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Roses</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
### The Term-document Matrix

**D1**: dogs are animals  
**D2**: cats are animals  
**D3**: orchids are plants  
**D4**: roses are plants

<table>
<thead>
<tr>
<th>Term</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cats</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dogs</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orchids</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Plants</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Roses</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Syntagmatic Relation Matrix

Top Syntagmatic Pairs

(animals, cats)
(animals, dogs)
(orchids, plants)
(plants, roses)

<table>
<thead>
<tr>
<th></th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cats</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dogs</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orchids</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Plants</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Roses</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Section 1

Basic Concepts and Theoretical Framework

• The Distributional Hypothesis

• Vector Space Models and the Term-Document Matrix

• Association Scores and Similarity Metrics

• The Curse of Dimensionality and Dimensionality Reduction

• Semantic Cognition, Conceptualization and Abstraction
Vector Space Models (VSMs)

- Vector Space Models have been extensively used in Artificial Intelligence and Machine Learning applications
- Vector Space Models for language applications were introduced by Gerard Salton* within the context of Information Retrieval
- Vector Spaces allow for simultaneously modeling words and the contexts in which they occur

Three Main VSM Constructs*

- The term-document matrix
  - Similarity of documents
  - Similarity of words (Syntagmatic Relations)
- The word-context matrix
  - Similarity of words (Paradigmatic Relations)
- The pair-pattern matrix
  - Similarity of relations

The Term-Document Matrix

- A model representing joint distributions between words and documents

| T₁ | T₂ | T₃ | T₄ | T₅ | T₆ | ... | D₁ | D₂ | D₃ | D₄ | D₅ | D₆ | D₇ | D₈ | D₉ | D₁₀ | D₁₁ | D₁₂ |
|-----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| v₁₁ | v₂₁ | v₃₁ | v₄₁ | v₅₁ | v₆₁ | ... | v₁₂ | v₂₂ | v₃₂ | v₄₂ | v₅₂ | v₆₂ | ... | v₁₉ | v₂₉ | v₃₉ | v₄₉ | v₅₉ |

Non-zero row values for those documents containing a given word

Non-zero column values for those words occurring within a given document
The Term-Document Matrix

- Each row of the matrix represents a unique vocabulary word in the data collection
- Each column of the matrix represents a unique document in the data collection
- Represents joint distributions between words and documents
- It is a bag-of-words kind of representation
- A real-valued weighting strategy is typically used to improve discriminative capabilities
A bag-of-words Type of Model

- Relative word orderings within the documents are not taken into account
Weighting Strategies

- More discriminative words are more important!

Zipf’s Law for Languages

- Very frequent words (function words)
- Frequent and infrequent words (content words)
- Very rare words (content words)
TF-IDF Weighting Scheme*

We want to favor words that are:

• Common within documents
  ▫ Term-Frequency Weight (TF): it counts how many times a word occurs within a document

• Uncommon across documents
  ▫ Inverse-Document-Frequency (IDF): it inversely accounts for the number of documents that contain a given word

TF-IDF Weighting Effects

Higher weights are given to those words that are frequent within but infrequent across documents.
TF-IDF Weighting Computation

- Term-Frequency (TF):
  \[ TF(w_i,d_j) = |w_i \in d_j| \]

- Inverse-Document-Frequency (IDF):
  \[ IDF(w_i) = \log \left( \frac{|D|}{1 + |d \in D : w_i \in d|} \right) \]

- TF-IDF with document length normalization:
  \[ TF-IDF(w_i,d_j) = \frac{TF(w_i,d_j) \cdot IDF(w_i)}{\sum_i |w_i \in d_j|} \]
PMI Weighting Scheme*

- Point-wise Mutual Information (PMI)
  \[ PMI(w_i,d_j) = \log \left( \frac{p(w_i,d_j)}{p(w_i)p(d_j)} \right) \]

- Positive PMI (PPMI)
  \[ PPMI(w_i,d_j) = \begin{cases} PMI(w_i,d_j) & \text{if } > 0 \\ 0 & \text{otherwise} \end{cases} \]

- Discounted PMI (compensates the tendency of PMI to increase the importance of infrequent events)
  \[ DPMI(w_i,d_j) = \delta_{ij} PMI(w_i,d_j) \]

Section 1

Basic Concepts and Theoretical Framework

• The Distributional Hypothesis
• Vector Space Models and the Term-Document Matrix
• Association Scores and Similarity Metrics
• The Curse of Dimensionality and Dimensionality Reduction
• Semantic Cognition, Conceptualization and Abstraction
Document Vector Spaces

Pay attention to the columns of the term-document matrix.

<table>
<thead>
<tr>
<th>variables</th>
<th>D_1</th>
<th>D_2</th>
<th>D_3</th>
<th>D_4</th>
<th>D_5</th>
<th>D_6</th>
<th>D_7</th>
<th>D_8</th>
<th>...</th>
<th>D_N</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Document Vector Spaces

Association scores and similarity metrics can be used to assess the degree of semantic relatedness among documents.
Word Vector Spaces

Pay attention to the rows of the term-document matrix

<table>
<thead>
<tr>
<th>observations</th>
<th>( D_1 )</th>
<th>( D_2 )</th>
<th>( D_3 )</th>
<th>( D_4 )</th>
<th>( D_5 )</th>
<th>( D_6 )</th>
<th>( D_7 )</th>
<th>( D_8 )</th>
<th>...</th>
<th>( D_N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_5 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_6 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_M )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

variables

term vector
Word Vector Spaces

Association scores and similarity metrics can be used to assess the degree of semantic relatedness among words.
Assessing Vector Similarities

- Association scores provide a means for measuring vector similarity
- Distances, on the other hand, provide a means for measuring vector dissimilarities
- Similarities and dissimilarities are in essence opposite measurements, and can be easily converted from one to another
Association Scores

- Dice: \( \text{dice}(V_1,V_2) = \frac{2 |N_1 \cap N_2|}{|N_1| + |N_2|} \)

- Jaccard: \( \text{jacc}(V_1,V_2) = \frac{|N_1 \cap N_2|}{|N_1 \cup N_2|} \)

- Cosine: \( \text{cos}(V_1,V_2) = \frac{\langle V_1,V_2 \rangle}{\|V_1\| \|V_2\|} \)
Distance Metrics

- **Hamming:** \( hm(V_1,V_2) = |N_1 \cap Z_2| + |Z_1 \cap N_2| \)

- **Euclidean:** \( d(V_1,V_2) = ||V_1 - V_2|| \)

- **citiblock:** \( cb(V_1,V_2) = ||V_1 - V_2||_1 \)

- **cosine:** \( dcos(V_1,V_2) = 1 - \cos(V_1,V_2) \)
Section 1

Basic Concepts and Theoretical Framework

• The Distributional Hypothesis
• Vector Space Models and the Term-Document Matrix
• Association Scores and Similarity Metrics
• The Curse of Dimensionality and Dimensionality Reduction
• Semantic Cognition, Conceptualization and Abstraction
The Curse of Dimensionality*

- Refers to the data sparseness problem that is intrinsic to high-dimensional spaces
- The problem results from the disproportionate increase of space volume with respect to the amount of available data
- If the statistical significance of results are to be maintained, then the amount of required data will grow exponentially with dimensionality

* Bellman, R.E. (1957), Dynamic programming, Princeton University Press
Dimensionality Reduction

- Deals with the “curse of dimensionality” problem
- Intends to explain the observations with less variables
- Attempts to find (or construct) the most informative variables

Provides a mathematical metaphor to the cognitive processes of Generalization and Abstraction!
Types of Dimensionality Reduction

Linear projections are like shadows

Non-linear projections preserve structure
Example of a Linear Projection

\[
\begin{array}{ccc}
A & B & C \\
X_A & X_B & X_C \\
Y_A & Y_B & Y_C \\
Z_A & Z_B & Z_C \\
\end{array}
\]

\[
\begin{array}{ccc}
A & B & C \\
W_A & W_B & W_C \\
\end{array}
\]
Example of a Non-linear Projection
The Case of Categorical Data

Set of Observations

<table>
<thead>
<tr>
<th></th>
<th>leaps</th>
<th>swims</th>
<th>eggs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frog</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Dolphin</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Kangaroo</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Shark</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Dissimilarity Matrix

<table>
<thead>
<tr>
<th></th>
<th>Frog</th>
<th>Dolp.</th>
<th>Kang.</th>
<th>Shark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frog</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Dolp.</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Kang.</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Shark</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Low-dimensional Embedding
Some Popular Methods

• Variable merging and pruning:
  ▫ Combine correlated variables (merging)
  ▫ Eliminate uninformative variables (pruning)

• Principal Component Analysis (PCA)
  ▫ Maximizes data variance in reduced space

• Multidimensional Scaling (MDS)
  ▫ Preserves data structure as much as possible

• Autoencoders
  ▫ Neural Network approach to Dimensionality Reduction
Variable Merging and Pruning

- Lemmatization and stemming (merging)
- Stop-word-list (pruning)

Term-Document Matrix after vocabulary merging and pruning
Principal Component Analysis (PCA)

- Eigenvalue decomposition of data covariance or correlation matrix (real symmetric matrix)

\[
M_{N \times N} = Q_{N \times N} \Lambda_{N \times N} Q_{N \times N}^T
\]

- Singular value decomposition (SVD) of data matrix

\[
M_{M \times N} = U_{M \times M} \Sigma_{M \times N} V_{N \times N}^T
\]
Latent Semantic Analysis (LSA)*

- Based on the Singular Value Decomposition (SVD) of a term-document matrix

\[ M_{M \times N} = U_{M \times M} \Sigma_{M \times N} V^T_{N \times N} \]

\[ \hat{M}_{M \times N} \approx U_{M \times K} \Sigma_{K \times K} V^T_{K \times N} \]

Multidimensional Scaling (MDS)

- Computes a low dimensional embedding by minimizing a “stress” function

\[
\text{Stress function} = \sqrt{\sum \sum (f(x_{ij}) - d_{ij})^2}
\]

- Metric MDS: directly minimizes stress function
- Non-metric MDS: relaxes the optimization problem by using a monotonic transformation
Autoencoders*

- Symmetric feed-forward non-recurrent neural network
  - Restricted Boltzmann Machine (pre-training)
  - Backpropagation (fine-tuning)

Section 1

Basic Concepts and Theoretical Framework

• The Distributional Hypothesis
• Vector Space Models and the Term-Document Matrix
• Association Scores and Similarity Metrics
• The Curse of Dimensionality and Dimensionality Reduction

• Semantic Cognition, Conceptualization and Abstraction
What is Cognition?

- Cognition is the process by which a sensory input is transformed, reduced, elaborated, stored, recovered, and used*

- Etymology:
  - Latin verb cognosco (“with”+“know”)
  - Greek verb gnósko (“knowledge”)

- It is a faculty that allows for processing information, reasoning and decision making

Three Important Concepts

• Memory: is the process in which information is encoded, stored, and retrieved

• Inference: is the process of deriving logical conclusions from premises known or assumed to be true (deduction, induction, abduction)

• Abstraction: is a generalization process by which concepts and rules are derived from a multiplicity of observations
Approaches to Semantic Cognition

• The hierarchical propositional approach*
  ▫ Concepts are organized in a hierarchical fashion

• The parallel distributed processing approach**
  ▫ Concept are stored in a distributed fashion and reconstructed by pattern completion mechanisms


Hierarchical Propositional Model

Example domain of living things
Advantages of Hierarchical Model

• Economy of storage

• Immediate generalization of
  ▫ known propositions to new members
  ▫ new propositions to known members

• Explains cognitive processes of *
  ▫ general-to-specific progression in children
  ▫ progressive deterioration in semantic dementia patients

Hierarchical Model Drawback!

There is strong experimental evidence of a graded category membership in human cognition

- Humans are faster verifying the statement *
  - ‘chicken is an animal’ than ‘chicken is a bird’
  - ‘robin is a bird’ than ‘chicken is a bird’

- This is better explained when the verification process is approached by means of assessing similarities across categories and elements

Parallel Distributed Processing*

- Semantic information is stored in a distributed manner across the system
- Semantic information is “reconstructed” by means of a pattern completion mechanism
- The reconstruction process is activated as the response to a given stimulus

Rumelhart Connectionist Network

Two-dimensional projection of the representation layer


Advantages of the PDP Model*

• Also explains both cognitive processes of development and degradation

• Additionally, it can explain the phenomenon of graded category membership:
  ▫ use of intermediate level categories (basic level**)
  ▫ over-generalization of more frequent items


PDP, DH and Vector Spaces

- The Parallel Distributed Processing (PDP) model explains a good amount of observed cognitive semantic phenomena.
- In addition, the connectionist approach has a strong foundation on neurophysiology.
- Both PDP and Distributional Hypothesis (DH) use differences/similarities over a feature space to model the semantic phenomenon.
- Vector Spaces constitute a great mathematical framework for this endeavor!!!
Section 1

Main references for this section

• M. Sahlgren, 2006, “The distributional hypothesis”

• P. D. Turney and P. Pantel, 2010, “From frequency to meaning: vector space models of semantics”


• G. Hinton and R. Salakhutdinov, 2006, “Reducing the dimensionality of data with neural networks”

Section 1

Additional references for this section


- Saussure, F. (1916) Course in General Linguistics


Section 1

Additional references for this section

- Bellman, R.E. (1957), Dynamic programming, Princeton University Press
Section 2

Vector Spaces in Monolingual NLP

- The Semantic Nature of Vector Spaces
- Information Retrieval and Relevance Ranking
- Word Spaces and Related Word Identification
- Semantic Compositionality in Vector Spaces
Constructing Semantic Maps

Document collection

"Semantic Map" of words or documents

Dimensionality Reduction

Vector Space of words or documents

TF-IDF Weighting

Vector Space of words or documents
Document Collection

- The Holy Bible
  - 66 books $\rightarrow$ 1189 chapters $\rightarrow$ 31103 verses
  - $\approx$700K running words $\rightarrow$ $\approx$12K vocabulary terms

*Distribution of verses per book within the collection*
Semantic Maps of Documents

Document collection

TF-IDF

Vector Space of documents

cosine distance

MDS

“Semantic Map” of documents

Dissimilarity Matrix
Semantic Maps of Documents

Old Testament

- Pentateuch
  - Historical books
    - Major prophets
      - Minor prophets
    - Wisdom books
  - Other Old Testament books

New Testament

- Gospels
  - Acts
  - Epistles (Paul)
- Epistles (others)

Revelation
Semantic Maps of Words

Document collection

TF-IDF

Vector Space of words

cosine distance

MDS

“Semantic Map” of words

Dissimilarity Matrix
Semantic Maps of Words

Non-living things

Living things

Sky

Land

Water
Discriminating Meta-categories

Opinionated content from rating website (Spanish)

- Positive and negative comments gathered from financial and automotive domains:
  - 2 topic categories: automotive and financial
  - 2 polarity categories: positive and negative
- Term-document matrix was constructed using full comments as documents
- A two-dimensional map was obtained by applying MDS to the vector space of documents
Discriminating Meta-categories
Section 2

Vector Spaces in Monolingual NLP

- The Semantic Nature of Vector Spaces
- Information Retrieval and Relevance Ranking
- Word Spaces and Related Word Identification
- Semantic Compositionality in Vector Spaces
Document Search: the IR Problem

- Given an informational need ("search query")
- and a very large collection of documents,
- find those documents that are relevant to it
Precision and Recall

How good a retrieval system is?

\[ TP = RD \cap SD \]
\[ TN = \neg RD \cap \neg SD \]
\[ FP = \neg RD \cap SD \]
\[ FN = RD \cap \neg SD \]

\[
\text{precision} = \frac{TP}{TP + FP} \quad \text{recall} = \frac{TP}{TP + FN} \quad F\text{-score} = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]
Binary Search

- Keyword based (query = list of keywords)
  - AND-search: selects documents containing all keywords in the query
  - OR-search: selects documents containing at least one of the keywords in the query
- Documents are either relevant or not relevant (binary relevance criterion)

Vector Space Search*

- Keyword based (query = list of keywords)
- Uses vector similarity scores to assess document relevance (a graded relevance criterion)

Precision/Recall Trade-off

Number of Selected Documents
(documents ranked according to vector similarity with the query)
Illustrative Example*

Consider a collection of 2349 paragraphs extracted from three different books:

- Oliver Twist by Charles Dickens
  - 840 paragraphs from 53 chapters
- Don Quixote by Miguel de Cervantes
  - 843 paragraphs from 126 chapters
- Pride and Prejudice by Jane Austen
  - 666 paragraphs from 61 chapters

Illustrative Example

Distribution of paragraphs per book and chapter

*Oliver Twist*

*Don Quixote*

Pride & Prejudice

Illustrative Example

Consider a set of 8 search queries:

<table>
<thead>
<tr>
<th>Query</th>
<th>Relevant Book and Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>oliver, twist, board</td>
<td>Oliver Twist, chapter 2</td>
</tr>
<tr>
<td>london, road</td>
<td>Oliver Twist, chapter 8</td>
</tr>
<tr>
<td>brownlow, grimwig, oliver</td>
<td>Oliver Twist, chapter 14</td>
</tr>
<tr>
<td>curate, barber, niece</td>
<td>Don Quixote, chapter 53</td>
</tr>
<tr>
<td>courage, lions</td>
<td>Don Quixote, chapter 69</td>
</tr>
<tr>
<td>arrival, clavileno, adventure</td>
<td>Don Quixote, chapter 93</td>
</tr>
<tr>
<td>darcy, dance</td>
<td>Pride &amp; Prejudice, chapter 18</td>
</tr>
<tr>
<td>gardiner, housekeeper, elizabeth</td>
<td>Pride &amp; Prejudice, chapter 43</td>
</tr>
</tbody>
</table>
Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary OR-search</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary AND-search</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vector@10 search</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Recall Bias**
- **Precision Bias**
- **F-score**
Automatic Relevance Feedback*

Use first search results to improve the search!

The most relevant documents should contain words that are good additional query keywords

The most irrelevant documents should contain words that are to be avoided as query keywords

newQuery = originalQuery + \alpha \frac{1}{|D_R|} \sum D_R - \beta \frac{1}{|D_{NR}|} \sum D_{NR}

Experimental Results

<table>
<thead>
<tr>
<th>Metric</th>
<th>Baseline</th>
<th>With ARF</th>
<th>Absolute Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Precision @10</td>
<td>30%</td>
<td>1.25%</td>
<td>1.25% abs gain</td>
</tr>
<tr>
<td>Mean Recall @10</td>
<td>20%</td>
<td>0.14%</td>
<td>0.14% abs gain</td>
</tr>
<tr>
<td>Mean F-score @10</td>
<td>10%</td>
<td>0.55%</td>
<td>0.55% abs gain</td>
</tr>
</tbody>
</table>
Section 2

Vector Spaces in Monolingual NLP

• The Semantic Nature of Vector Spaces
• Information Retrieval and Relevance Ranking
• Word Spaces and Related Word Identification
• Semantic Compositionality in Vector Spaces
Latent Semantic Analysis (LSA)

Document collection

Better Semantic Properties

Reduced-dimensionality Space

Vector Space Model
Latent Semantic Analysis (LSA)*

SVD: $M_{M\times N} = U_{M\times M} \Sigma_{M\times N} V_{N\times N}^T$

$U_{M\times M}^T M_{M\times N} = D_{M\times N}$

Documents projected into word space

$U_{K\times M}^T M_{M\times N} = D_{K\times N}$

Documents projected into reduced word space

$M_{M\times N} V_{N\times K} = W_{M\times K}$

Words projected into reduced document space

$V_{N\times K}^T = V_{N\times N}$

Words projected into document space

Dataset Under Consideration*

Term definitions from Spanish dictionary used as documents

<table>
<thead>
<tr>
<th>Collection</th>
<th>Terms</th>
<th>Definitions</th>
<th>Aver. Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verbs</td>
<td>4,800</td>
<td>12,414</td>
<td>6.05 words</td>
</tr>
<tr>
<td>Adjectives</td>
<td>5,390</td>
<td>8,596</td>
<td>6.05 words</td>
</tr>
<tr>
<td>Nouns</td>
<td>20,592</td>
<td>38,689</td>
<td>9.56 words</td>
</tr>
<tr>
<td>Others</td>
<td>5,273</td>
<td>9,835</td>
<td>8.01 words</td>
</tr>
<tr>
<td>Complete</td>
<td>36,055</td>
<td>69,534</td>
<td>8.32 words</td>
</tr>
</tbody>
</table>

- A document vector space for “verbs” is constructed
- LSA is used to project into a latent semantic space
- MDS is used to create a 2D map for visualization purposes

Differentiating Semantic Categories

Two semantic categories of verbs are considered

<table>
<thead>
<tr>
<th>Group A</th>
<th>Group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ayudar (to help)</td>
<td>Agredir (to threaten)</td>
</tr>
<tr>
<td>Compartir (to share)</td>
<td>Destruir (to destroy)</td>
</tr>
<tr>
<td>Beneficiar (to benefit)</td>
<td>Aniquilar (to eliminate)</td>
</tr>
<tr>
<td>Colaborar (to collaborate)</td>
<td>Atacar (to attack)</td>
</tr>
<tr>
<td>Salvar (to save)</td>
<td>Arruinar (to ruin)</td>
</tr>
<tr>
<td>Apoyar (to support)</td>
<td>Matar (to kill)</td>
</tr>
<tr>
<td>Cooperar (to cooperate)</td>
<td>Perjudicar (to perjudice)</td>
</tr>
<tr>
<td>Favorecer (to favour)</td>
<td>—</td>
</tr>
</tbody>
</table>
Differentiating Semantic Categories

No LSA applied: original dimensionality maintained
Differentiating Semantic Categories

LSA used to project into latent space of 800 dimensions
Differentiating Semantic Categories

LSA used to project into latent space of 400 dimensions
Differentiating Semantic Categories

LSA used to project into latent space of 100 dimensions
Semantic Similarity of Words

The totality of the 12,414 entries for verbs were considered

- An 800-dimensional latent space representation was generated by applying LSA
- k-means was applied to group the 12,414 entries into 1,000 clusters (minimum size 2, maximum size 36, mean size 12.4, variance 4.7)
- Finally, non-linear dimensionality reduction (MDS) was applied to generate a map
Semantic Similarity of Words

- to put under the sun
- to laugh
- to study
- to write
- to read
- to cry
- to jump
- to walk
- to sail
- to swim
- to water
- to raise crops
- to rain
Regularities in Vector Spaces*

Recurrent Neural Network Language Model

• After study internal word representations generated by the model

• Syntactic and semantic regularities were discovered to be mapped into the form of constant vector offsets

Recurrent Neural Network (RNN)

\[ h(t) = \text{Sigmoid}(W x(t) + R h(t-1)) \]

\[ y(t) = \text{Softmax}(V h(t)) \]
Regularities as Vector Offsets

Kings – King ≈ Queens – Queen

Queens ≈ Kings – King + Queen

Gender offset

Singular/plural offset

Comparative Evaluations

Propositions formulated as analogy questions:
“x is to y as m is to ___”

Syntactic Evaluation (8000 propositions)*

- LSA-320: 17%
- RNN-320: 29%

Semantic Evaluation (79 propositions from SemEval 2012)**

- LSA-320: 36%
- RNN-320: 40%


Section 2

Vector Spaces in Monolingual NLP

- The Semantic Nature of Vector Spaces
- Information Retrieval and Relevance Ranking
- Word Spaces and Related Word Identification
- **Semantic Compositionality in Vector Spaces**
Semantic Compositionality

• The *principle of compositionality* states that the meaning of a complex expression depends on:
  ▫ the meaning of its constituent expressions
  ▫ the rules used to combine them

• Some idiomatic expressions and named entities constitute typical exceptions to the principle of compositionality in natural language
Compositionality and Exceptions

Consider the adjective-noun constructions

RED CAR

WHITE HOUSE

???
Compositionality in Vector Space

- *Can this principle be modeled in Vector Space representations of language?*

- Two Basic mechanisms can be used to model compositionality in the vector space model framework*
  - Intersection of properties (multiplicative approach)
  - Combination of properties (additive approach)

Compositionality Models

• Given two word vector representations \( x \) and \( y \)

• A composition vector \( z \) can be computed as:

\[
\begin{align*}
\text{Multiplicative Models} & : & z &= C \times x \times y \\
& & z_i &= \sum_j x_j y_{i-j} \\
& & z_i &= x_i y_i \\
\text{Additive Models} & : & z &= A \times x + B \times y \\
& & z_i &= x_i + y_i \\
& & z_i &= \alpha x_i + \beta y_i \\
& & z_i &= \alpha x_i + \beta y_i + \gamma x_i y_i
\end{align*}
\]
Additive Compositionality*

- Use unigram and bigram counts to identify phrases
- Uses Skip-gram model to compute word representations
- Compute element-wise additions of word vectors to retrieve associated words:
  - Czech + currency → koruna, Check crown, ...
  - German + airline → airline Lufthansa, Lufthansa, ...
  - Russian + river → Moscow, Volga River, ...

Adjectives as Linear Maps*

- An adjective-noun composition vector is: \( z = A n \)
- The rows of \( A \) are estimated by linear regressions
- Some examples of predicted nearest neighbors:
  - general question  \( \rightarrow \) general issue
  - recent request  \( \rightarrow \) recent enquiry
  - current dimension  \( \rightarrow \) current element
  - special something  \( \rightarrow \) special thing

*Baroni M. and Zamparelli R. (2010), Nous are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space, in EMNLP 2010
Section 2

Main references for this section

• G. Salton, A. Wong and C. S. Yang, 1975, “A Vector Space for Automatic Indexing”

• R. E. Banchs, 2013, “Text Mining with MATLAB”


Section 2

Additional references for this section


- Baroni M. and Zamparelli R. (2010), Nous are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space, in EMNLP 2010
Section 3

Vector Spaces in Cross-language NLP

• **Semantic Map Similarities Across Languages**
• Cross-language Information Retrieval in Vector Spaces
• Cross-script Information Retrieval and Transliteration
• Cross-language Sentence Matching and its Applications
• Semantic Context Modelling for Machine Translation
• Bilingual Dictionary and Translation-table Generation
• Evaluating Machine Translation in Vector Space
Semantic Maps Revisited

Document collection

“Semantic Map” of documents

Vector Space of documents

Dissimilarity Matrix

TF-IDF

MDS

cosine distance
Multilingual Document Collection

66 Books from The Holy Bible: English version

(vocabulary size: 8121 words)
Multilingual Document Collection

66 Books from The Holy Bible: Chinese version

(vocabulary size: 12952 words)
Multilingual Document Collection

66 Books from The Holy Bible: Spanish version

(vocabulary size: 25385 words)
Cross-language Similarities

- Each language map has been obtained independently from each other language (monolingual context).
- The similarities among the maps are remarkable.
- *Could we exploit these similarities for performing cross-language information retrieval tasks?*
Section 3

Vector Spaces in Cross-language NLP

- Semantic Map Similarities Across Languages
- **Cross-language Information Retrieval in Vector Spaces**
- Cross-script Information Retrieval and Transliteration
- Cross-language Sentence Matching and its Applications
- Semantic Context Modelling for Machine Translation
- Bilingual Dictionary and Translation-table Generation
- Evaluating Machine Translation in Vector Space
Semantic Maps for CLIR
CLIR by Using MDS Projections*

• Start from a multilingual collection of “anchor documents” and construct the retrieval map
• Project new documents and queries from any source language into the retrieval language map
• Retrieve documents over retrieval language map by using a distance metric

CLIR by Using MDS Projections

Source Language Vector Space

Retrieval Language Vector Space

Anchor Documents

Query placement

New document placement

MDS

Retrieval Map
Computing a Projection Matrix

A linear transformation from the original high dimensional space into the lower dimensionality map can be inferred from anchor documents

\[ M = T D \]

Coordinates of anchor documents in the projected space \((K \times N)\)

Distances among anchor documents in the original space \((N \times N)\)

Transformation Matrix \((K \times N)\)

\[ T = M D^{-1} \]
Projecting Documents and Queries

A probe document or query can be placed into the retrieval map by using the transformation matrix

\[ m = T d \]

- **Transformation Matrix** \((K \times N)\)
- **Coordinates of probe document (or query) in the projected space of retrieval language**
- **Distances between probe document (or query) and anchor documents in the original language space**
Computing a Projection Matrix

Two different variants of the linear projection matrix $T$ can be computed:

- A monolingual projection matrix:*
  - $M$ and $D$ are computed on the retrieval language

- A cross-language projection matrix:**
  - $M$ is computed on the retrieval language, and
  - $D$ is computed on the source language


Monolingual Projection Method

\[ m = (MD^{-1})d \]

Monolingual projection matrix

Source language

Retrieval language

D

MDS

Retrieval map
Cross-language Projection Method

\[ m = (MD^{-1}) d \]

Cross-language projection matrix

Source language

Retrieval language

MDS

Retrieval map
CLIR by Using Cross-language LSI*

- In monolingual LSI, the term-document matrix is decomposed into a set of $K$ orthogonal factors by means of Singular Value Decomposition (SVD).

- In cross-language LSI, a multilingual term-document matrix is constructed from a multilingual parallel collection and LSI is applied by considering multilingual “extended” representations of query and documents.

The Cross-language LSI Method

\[ X = \begin{pmatrix} X_a \\ X_b \end{pmatrix} \]

**SVD:** \[ X = U \Sigma V^T \]

Retrieval is based on internal product of the form:

\[ \langle U^T d, U^T q \rangle \]

With:

\[ d = \begin{pmatrix} d_a \\ o \end{pmatrix} \text{ or } \begin{pmatrix} o \\ d_b \end{pmatrix} \]

\[ q = \begin{pmatrix} q_a \\ o \end{pmatrix} \text{ or } \begin{pmatrix} o \\ q_b \end{pmatrix} \]

Multilingual term-document matrix

Term-document matrix in language A

Term-document matrix in language B
Comparative Evaluations

We performed a comparative evaluation of the three methods described over the trilingual dataset:

• Task 1: Retrieve a book using the same book in a different language as query:
  ▫ Subtask 1.A: Dimensionality of the retrieval space is varied
  ▫ Subtask 1.B: Anchor document set size is varied

• Task 2: Retrieve a chapter using the same chapter in a different language as a query
Task 1.A: Dimensionality of Space

Retrieval carried out over Chinese Map

English to Chinese
Task 1.B: Anchor Document Set

Retrieval carried out over Chinese Map

English to Chinese (dimensionality of retrieval space is equal to anchor set size)
Task 2: Chapter Retrieval

Retrieval carried out over Chinese Map

English to Chinese
(dimensionality of retrieval space is equal to anchor set size)
Some Conclusions*

- Semantic maps, and more specifically MDS projections, can be exploited for CLIR tasks
- The cross-language projection matrix variant performs better than the monolingual projection matrix variant
- MDS maps perform better than LSI for the considered CLIR tasks

Section 3

Vector Spaces in Cross-language NLP

- Semantic Map Similarities Across Languages
- Cross-language Information Retrieval in Vector Spaces
- Cross-script Information Retrieval and Transliteration
- Cross-language Sentence Matching and its Applications
- Semantic Context Modelling for Machine Translation
- Bilingual Dictionary and Translation-table Generation
- Evaluating Machine Translation in Vector Space
Main Scripts used Around the World

- Amharic alphabet
- Arabic alphabet
- Armenian alphabet
- Bengali alphabet
- Burmese alphabet
- Chinese script
- Cyrillic alphabet
- Devanagari alphabet
- Georgian alphabet
- Greek alphabet
- Hebrew alphabet
- Japanese script
- Khmer alphabet
- Korean alphabet
- Lao alphabet
- Latin alphabet
- Sinhala alphabet
- Thai alphabet
- Tibetan alphabet
Transliteration and Romanization

- The process of phonetically representing the words of one language in a non-native script
- Due to socio-cultural and technical reasons, most languages using non Latin native scripts commonly implement Latin script writing rules: “Romanization”

你好 ➡️ nǐ hǎo
The Multi-Script IR (MSIR) Problem*

- There are many languages that use non Latin scripts (Japanese, Chinese, Arabic, Hindi, etc.)
- There is a lot of text for these languages in the Web that is represented into the Latin script
- For some of these languages, no standard rules exist for transliteration

The Main Challenge of MSIR

- Mixed script queries and documents
- Extensive spelling variations

Native Script

Non-native Script

Mixed Script

Spelling variations
Significance of MSIR

- Only 6% of the queries issued in India to Bing contain Hindi words in Latin script
- From a total number of 13.78 billion queries!!!

800 million queries!!!

*People (6%)*
*Organizations (14%)*
*Locations (8%)*
*Movies (7%)*
*Songs & lyrics (18%)*
*Websites (22%)*
*others (25%)*
Proposed Method for MSIR*

- Use characters and bigram of characters as terms (features) and words as documents (observations)
- Build a cross-script semantic space by means of a deep autoencoder
- Use the cross-script semantic space for finding “equivalent words” within and across scripts
- Use “equivalent words” for query expansion

Training the Deep Autoencoder

Building the Semantic Space

2D Visualization of the constructed cross-script semantic space

[Native Script | ooo00...0]
[0000000...0 | Latin Script]

Cross-script query expansion

<table>
<thead>
<tr>
<th>Original Query</th>
<th>ik din ayega</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Variants of “ik”</td>
<td>“ik”, “ek”, “एक”</td>
</tr>
<tr>
<td>Variants of “din”</td>
<td>“din”, “diin”, “दिन”</td>
</tr>
<tr>
<td>Variants of “ayega”</td>
<td>“ayega”, “aeyega”, “ayega”, “आयेगा”</td>
</tr>
</tbody>
</table>
| Formulated Query (bigram) | ik$din, ik$diin, ⋯
                               | diin$ayegaa, “एक$दिन”,
                               | “दिन$आयेगा” |
Baseline Systems

The proposed method is compared to:

• Naïve system: no query expansion used

• LSI: uses cross-language LSI to find the word equivalents

• CCA: uses Canonical Correlation Analysis* to find the word equivalents

## Comparative Evaluation Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Average Precision</th>
<th>Similarity Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve</td>
<td>29.10%</td>
<td>NA</td>
</tr>
<tr>
<td>LSI</td>
<td>35.22%</td>
<td>0.920</td>
</tr>
<tr>
<td>CCA</td>
<td>38.91%</td>
<td>0.997</td>
</tr>
<tr>
<td>Autoencoder</td>
<td>50.39%</td>
<td>0.960</td>
</tr>
</tbody>
</table>
Number of “Word Equivalents”

Section 3

Vector Spaces in Cross-language NLP

- Semantic Map Similarities Across Languages
- Cross-language Information Retrieval in Vector Spaces
- Cross-script Information Retrieval and Transliteration
- Cross-language Sentence Matching and its Applications
- Semantic Context Modelling for Machine Translation
- Bilingual Dictionary and Translation-table Generation
- Evaluating Machine Translation in Vector Space
Cross-language Sentence Matching

• Focuses on the specific problem of text matching at the sentence level

• A segment of text in a given language is used as a query for retrieving a similar segment of text in a different language

• This task is useful to some specific applications:
  ▫ Parallel corpora compilation
  ▫ Cross-language plagiarism detection
Parallel Corpora Compilation*

• Deals with the problem of extracting parallel sentence from comparable corpora

CL Plagiarism Detection*

- Deals with the problem of identifying copied documents or fragments across languages

*Potthast M., Stein B., Eiselt A., Barrón A. and Rosso P. (2009), Overview of the 1st international competition on plagiarism detection, Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse
Proposed Method

• The previously described MDS-based Semantic Map approach to CLIR is used
  ▫ Cross-language projection matrix variant*
  ▫ Additionally, a majority voting strategy over different semantic retrieval maps is implemented and tested

Majority Voting Strategy

Retrieval Map 1

Retrieval Map 2

Retrieval Map K

Global Ranking

d2
d1
d3

Ranking 1

d1
d2
d3

Ranking 2

d2
d1
d3

Ranking K

d2
d1
d3
# Penta-lingual Data Collection

*Extracted from the Spanish Constitution*

<table>
<thead>
<tr>
<th>Language</th>
<th>Sample sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>This right may not be restricted for political or ideological reasons</td>
</tr>
<tr>
<td>Spanish</td>
<td>Este derecho no podrá ser limitado por motivos políticos o ideológicos</td>
</tr>
<tr>
<td>Català</td>
<td>Aquest dret no podrà ser limitat por motius polítics o ideològics</td>
</tr>
<tr>
<td>Euskera</td>
<td>Eskubide hau arrazoi politiko edo idiólogikoek ezin dute mugatu</td>
</tr>
<tr>
<td>Galego</td>
<td>Este dereito non poderá ser limitado por motivos políticos ou ideolóxicos</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>English</th>
<th>Spanish</th>
<th>Català</th>
<th>Euskera</th>
<th>Galego</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sentences</td>
<td>611</td>
<td>611</td>
<td>611</td>
<td>611</td>
<td>611</td>
</tr>
<tr>
<td>Number of words</td>
<td>15285</td>
<td>14807</td>
<td>15423</td>
<td>10483</td>
<td>13760</td>
</tr>
<tr>
<td>Vocabulary size</td>
<td>2080</td>
<td>2516</td>
<td>2523</td>
<td>3633</td>
<td>2667</td>
</tr>
<tr>
<td>Average sentence length</td>
<td>25.01</td>
<td>24.23</td>
<td>25.24</td>
<td>17.16</td>
<td>22.52</td>
</tr>
</tbody>
</table>
Task Description

• To retrieve a sentence from the English version of the Spanish Constitution using the same sentence in any of the other four languages as a query

• Performance quality is evaluated by means of top-1 and top-5 accuracies measured over a 200-sentence test set

• One retrieval map is constructed for each language available in the collection (400 anchor documents)

• Retrieval Map dimensionality for all languages: 350
## Evaluation Results

<table>
<thead>
<tr>
<th>Retrieval Map</th>
<th>Spanish</th>
<th>Català</th>
<th>Euskera</th>
<th>Galego</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>top-1</strong></td>
<td>97.0</td>
<td>95.5</td>
<td>96.5</td>
<td>96.5</td>
</tr>
<tr>
<td><strong>top-5</strong></td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>

| English       | 97.0    | 95.5   | 77.0    | 76.0   |
| Spanish       | 95.5    | 99.0   | 94.5    | 94.5   |
| Català        | 95.0    | **100**| 94.5    | 94.5   |
| Euskera       | 96.5    | 99.0   | 95.0    | 95.0   |
| Galego        | 96.5    | 99.0   | 95.0    | 93.0   |
| Majority voting | 97.5 | **100**| 96.5    | **100**|

| top-1         | 95.0    | 94.0   | 93.0    | 93.0   |
| top-5         | 98.5    | 99.5   | 99.0    | 98.0   |
Comparative Evaluation

• The proposed method (majority voting result) is compared to other two methods:
  ▫ Cross-language LSI* (previously described)
  ▫ Query translation** (a cascade combination of machine translation and monolingual information retrieval)


** Chen J. and Bao Y. (2009), Cross-language search: The case of Google language tools, First Monday, 14(3-2)
## Comparative Evaluation Results

<table>
<thead>
<tr>
<th></th>
<th>Spanish</th>
<th>Català</th>
<th>Euskera</th>
<th>Galego</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLIR Method</strong></td>
<td>top-1</td>
<td>top-5</td>
<td>top-1</td>
<td>top-5</td>
</tr>
<tr>
<td>LSI based</td>
<td>96.0</td>
<td>99.0</td>
<td>95.5</td>
<td>98.5</td>
</tr>
<tr>
<td>Query transl.</td>
<td>96.0</td>
<td>99.0</td>
<td>95.5</td>
<td>99.5</td>
</tr>
<tr>
<td>Semantic maps</td>
<td>97.5</td>
<td>100</td>
<td>96.5</td>
<td>99.5</td>
</tr>
</tbody>
</table>

*Euskera-to-English* translations were not available
Section 3

Vector Spaces in Cross-language NLP

- Semantic Map Similarities Across Languages
- Cross-language Information Retrieval in Vector Spaces
- Cross-script Information Retrieval and Transliteration
- Cross-language Sentence Matching and its Applications
- **Semantic Context Modelling for Machine Translation**
- Bilingual Dictionary and Translation-table Generation
- Evaluating Machine Translation in Vector Space
Statistical Machine Translation

Developing context-awareness in SMT systems

• Original noisy channel formulation:

\[ T = \arg \max_T P(T|S) = \arg \max_T P(S|T) P(T) \]

• Proposed model reformulation*:

\[ T = \arg \max_T P(T|S,C) = \arg \max_T P(C|S,T) P(S|T) P(T) \]

Unit Selection Depends on Context

**S1:** the murderer shall be put to death by the mouth of witnesses

*por el testimonio de* testigos se dará muerte al asesino

**S2:** roll great stones upon the mouth of the cave

*haced rodar grandes piedras a la entrada de la cueva*

**S3:** to fulfill the word of the Lord by the mouth of Jeremiah

*para que se cumpliese la palabra de Jehovah por la boca de Jeremías*

**S4:** then Nebuchadnezzar came near to the mouth of the burning fiery furnace

*entonces Nabucodonosor se acercó a la puerta del horno de fuego ardiendo*

**Input:** but by every word that proceeded out of the mouth of the Lord doth man live
An Actual Example...

“WINE” sense of “VINO”

SC1: No habéis comido pan ni tomado vino ni licor...
Ye have not eaten bread, neither have ye drunk wine or strong drink...

SC2: ...dieron muchas primicias de grano, vino nuevo, aceite, miel y de todos ...
... brought in abundance the first fruits of corn, wine, oil, honey, and of all ...

“CAME” sense of “VINO”

SC3: Al tercer día vino Jeroboam con todo el pueblo a Roboam ...
So Jeroboam and all the people came to Rehoboam the third day ...

SC4: Ella vino y ha estado desde la mañana hasta ahora ...
She came, and hath continued even from the morning until now ...

IN1: ... una tierra como la vuestra, tierra de grano y de vino, tierra de pan y de viñas ...
IN2: Cuando amanecía, la mujer vino y cayó delante de la puerta de la casa de aquel ...

(came)
Translation probabilities

• Translation probabilities:

| Phrase                | \( \phi(f|e) \) | \( \text{lex}(f|e) \) | \( \phi(e|f) \) | \( \text{lex}(e|f) \) |
|----------------------|-----------------|-----------------|-----------------|-----------------|
| \{vino|||wine\}      | 0.665198        | 0.721612        | 0.273551        | 0.329431        |
| \{vino|||came\}      | 0.253568        | 0.131398        | 0.418478        | 0.446488        |

• Proposed context-awareness model:

<table>
<thead>
<tr>
<th></th>
<th>SC1</th>
<th>SC2</th>
<th>SC3</th>
<th>SC4</th>
</tr>
</thead>
<tbody>
<tr>
<td>sense</td>
<td>{vino</td>
<td></td>
<td></td>
<td>wine}</td>
</tr>
<tr>
<td>IN1</td>
<td>0.0636</td>
<td>0.2666</td>
<td>0.0351</td>
<td>0.0310</td>
</tr>
<tr>
<td>IN2</td>
<td>0.0023</td>
<td>0.0513</td>
<td>0.0888</td>
<td>0.0774</td>
</tr>
</tbody>
</table>
### Comparative evaluation*

<table>
<thead>
<tr>
<th>Model</th>
<th>Development</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline System</td>
<td>39.92</td>
<td>38.92</td>
</tr>
<tr>
<td>Vector Space Model</td>
<td>40.61</td>
<td>39.43</td>
</tr>
<tr>
<td>Statistical Class Model</td>
<td>40.62</td>
<td>39.72</td>
</tr>
<tr>
<td>Latent Dirichlet Allocation</td>
<td>40.63</td>
<td>39.82</td>
</tr>
<tr>
<td>Latent Semantic Indexing</td>
<td>40.80</td>
<td>39.86</td>
</tr>
</tbody>
</table>

Neural Network Models for MT*

- The Neural Network framework can be used to incorporate source context information in both:
  - the target language model:
    \[ \text{Neural Network Joint Model (NNJM)} \]
  - the translation model:
    \[ \text{Neural NetworkLexical Translation Model (NNLTM)} \]

Joint Model (NNJM)

- Estimates the probability of a target word given its previous word history and a source context window

\[
P(T|S) \approx \prod_{i=1}^{\lfloor T \rfloor} P( t_i | t_{i-1}, t_{i-2} \ldots t_{i-n}, s_{j+m}, s_{j+m-1} \ldots s_j \ldots s_{j-m+1}, s_{j-m} )
\]
Lexical Translation Model (NNLTM)

- Estimates the probability of a target word given a source context window

\[
P(T|S) \approx \prod_{j=1}^{|S|} P(t_i | s_{j+m}, s_{j+m-1}, \ldots, s_j, \ldots, s_{j-m+1}, s_{j-m})
\]

with \( i = f_a(j) \)
Neural Network Architecture

• Feed-forward Neural Network Language Model

\[ y = V f( b + W [C w_{t-1}, C w_{t-2} ... C w_{t-n}] ) \]

\[ y_i = p(w_t = i \mid \text{context}) \]

Experimental Results*

Section 3

Vector Spaces in Cross-language NLP

- Semantic Map Similarities Across Languages
- Cross-language Information Retrieval in Vector Spaces
- Cross-script Information Retrieval and Transliteration
- Cross-language Sentence Matching and its Applications
- Semantic Context Modelling for Machine Translation
- **Bilingual Dictionary and Translation-table Generation**
- Evaluating Machine Translation in Vector Space
Word Translations in Vector Space

- Semantic similarities across languages can be exploited to “discover” word translation pairs from parallel data collections by:
  - either operating in the term-document matrix space*
  - or learning transformations across reduced spaces**


Operating in Term-document Space*

Parallel corpus (aligned at sentence level)

Term-document matrix (Spanish)

Term-document matrix (English)

Vectors of parallel documents associated to term \( w \)

Vectors of parallel documents dissociated to term \( w \)

Obtaining the Translation Terms*

- Compute $V^+$, the average vector of parallel documents associated to term $w$
- Compute $V^-$, the average vector of parallel documents dissociated to term $w$
- Obtain the most relevant terms (with largest weights) for the difference vector $V^+ - V^-$

Some Sample Translations

- **English translations to Spanish terms:**
  - casa: house, home
  - ladrón: thief, sure, fool
  - caballo: horse, horseback

- **Spanish translations to English terms:**
  - city: ciudad, fortaleza
  - fields: campo, vida
  - heart: corazón, ánimo, alma
Learning Projections*

- Construct projection spaces by means of
  - either CBOW model (Continuous Bag-Of-Words)
  - or Skip-gram model

Some Sample Projections

English Semantic Map for Animals

Spanish Semantic Map for Animals

Obtaining the Translation Terms

- Use some bilingual word pairs \( \{s_i, t_i\} \) to train a “translation matrix” \( W \) such that:
  \[
  t_i \approx W s_i
  \]
- Use \( W \) for projecting a new term \( s_j \) into the target space
- Collect the terms in target space that are closest to the obtained projection
Some Sample Translations*

- English translations to Spanish terms:
  - emociones: emotions, emotion, feeling
  - imperio: dictatorship, imperialism, tyranny
  - preparada: prepared, ready, prepare
  - millas: kilometers, kilometres, miles
  - hablamos: talking, talked, talk

The BI-CVM Model*

Compositional Sentence Model

\[ a_{\text{root}} = \sum_{i=0}^{\mid a \mid} a_i \]

Objective Function

Minimizes:
\[ E_{\text{dist}}(a,b) = \| a_{\text{root}} - b_{\text{root}} \|^2 \]

Maximizes:
\[ E_{\text{dist}}(a,n) = \| a_{\text{root}} - n_{\text{root}} \|^2 \]

Non Parallel Sentences (randomly selected)

Some Sample Projections

Days of the Week

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td>lundi</td>
<td>montag</td>
</tr>
<tr>
<td>Tuesday</td>
<td>mardi</td>
<td>dientag</td>
</tr>
<tr>
<td>Wednesday</td>
<td>mercredi</td>
<td>mercredi</td>
</tr>
<tr>
<td>Thursday</td>
<td>jeudi</td>
<td>donnerstag</td>
</tr>
<tr>
<td>Friday</td>
<td>vendredi</td>
<td>freitag</td>
</tr>
<tr>
<td>Saturday</td>
<td>samedi</td>
<td>samstag</td>
</tr>
</tbody>
</table>

Months of the Year

<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>janvier</td>
<td>januar</td>
</tr>
<tr>
<td>February</td>
<td>février</td>
<td>februar</td>
</tr>
<tr>
<td>March</td>
<td>mars</td>
<td>mars</td>
</tr>
<tr>
<td>April</td>
<td>avril</td>
<td>april</td>
</tr>
<tr>
<td>May</td>
<td>mai</td>
<td>mai</td>
</tr>
<tr>
<td>June</td>
<td>juin</td>
<td>juni</td>
</tr>
<tr>
<td>July</td>
<td>juillet</td>
<td>juillet</td>
</tr>
<tr>
<td>August</td>
<td>août</td>
<td>aout</td>
</tr>
<tr>
<td>September</td>
<td>septembre</td>
<td>septembre</td>
</tr>
<tr>
<td>October</td>
<td>octobre</td>
<td>oktober</td>
</tr>
<tr>
<td>November</td>
<td>novembre</td>
<td>novembre</td>
</tr>
<tr>
<td>December</td>
<td>décembre</td>
<td>dezember</td>
</tr>
</tbody>
</table>

Section 3

Vector Spaces in Cross-language NLP

- Semantic Map Similarities Across Languages
- Cross-language Information Retrieval in Vector Spaces
- Cross-script Information Retrieval and Transliteration
- Cross-language Sentence Matching and its Applications
- Semantic Context Modelling for Machine Translation
- Bilingual Dictionary and Translation-table Generation
- **Evaluating Machine Translation in Vector Space**
Automatic Evaluation of MT

ASR

MT

output

transcription

reference

output

UNIQUE

NON

UNIQUE
Human Evaluation of MT*

ADEQUACY
How much of the source information is preserved?

FLUENCY
How good is the generated target language quality?

\[ P(T|S) \approx P(S|T) \cdot P(T) \]

Proposed Evaluation Framework*

• Approximate adequacy and fluency by means of independent models:
  ▫ Use a “semantic approach” for adequacy
  ▫ Use a “syntactic approach” for fluency

• Combine both evaluation metrics into a single evaluation score

AM: Adequacy-oriented Metric

• Compare sentences in a semantic space
  ▫ Monolingual AM (mAM): compare output vs. reference
  ▫ Cross-language AM (xAM): compare output vs. input
FM: Fluency-oriented Metric

- Measures the quality of the target language with a language model
- Uses a compensation factor to avoid effects derived from differences in sentence lengths
AM-FM Combined Score

Both components can be combined into a single metric according to different criteria

- Weighted Harmonic Mean: \( H-AM-FM = \frac{AM \cdot FM}{\alpha AM + (1-\alpha) FM} \)

- Weighted Mean: \( M-AM-FM = (1-\alpha) AM + \alpha FM \)

- Weighted L2-norm: \( N-AM-FM = \sqrt{(1-\alpha) AM^2 + \alpha FM^2} \)
WMT-2007 Dataset*

- Fourteen tasks:
  - five European languages (EN, ES, DE, FR, CZ) and
  - two different domains (News and EPPS).
- Systems outputs available for fourteen of the fifteen systems that participated in the evaluation.
- 86 system outputs for a total of 172,315 individual sentence translations, from which 10,754 were rated for both adequacy and fluency by human judges.

Dimensionality Selection

Pearson’s correlation coefficients between the $mAM$ (left) and $xAM$ (right) components and human-generated scores for adequacy.
mAM-FM and Adequacy
mAM-FM and Fluency
xAM-FM and Adequacy
xAM-FM and Fluency

![Graph showing correlation coefficient with fluency against weighting parameter α. The graph compares different weighting schemes such as Weighted L2-Norm, Weighted Mean, Weighted Harmonic Mean, BLEU, and Meteor.]
Section 3

Main references for this section

Section 3

Main references for this section


Section 3

Additional references for this section


• Potthast M., Stein B., Eiselt A., Barrón A. and Rosso P. (2009), Overview of the 1st international competition on plagiarism detection, Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse
Section 3

Additional references for this section

- Chen J. and Bao Y. (2009), Cross-language search: The case of Google language tools, First Monday, 14(3-2)
Section 4

Future Research and Applications

• Current limitations of vector space models
• Encoding word position information into vectors
• From vectors and matrices to tensors
• Final remarks and conclusions
Conceptual vs. Functional

- Vector Space Models are very good to capture the conceptual aspect of meaning
  - \{dog, cow, fish, bird\} vs. \{chair, table, sofa, bed\}

- However, they still fail to properly model the functional aspect of meaning
  - “Give me a pencil” vs. “Give me that pencil”
Word Order Information Ignored

• Differently from Formal Semantics*, VSM lacks of a clean interconnection between the syntax and semantic phenomena

• In part, a consequence of the Bag-Of-Words nature of VSM

VSMs completely ignore word order information

Non-unique Representations

• Consider the two following sentences*
  ▫ “That day the office manager, who was drinking, hit the problem sales worker with a bottle, but it was not serious”
  ▫ “It was not the sales manager, who hit the bottle that day, but the office worker with a serious drinking problem”

• Although they are completely different, they contain exactly the same set of words, so they will produce exactly the same VSM representation!

Other Limitations

*Additionally...*

- VSMs are strongly data-dependent
- VSMs noisy in nature (spurious events)
- Uncertainty or confidence estimation becomes an important issue
- Multiplicity of parameters with not clear relation to the outcomes
Section 4

Future Research and Applications

• Current limitations of vector space models
• **Encoding word position information into vectors**
• From vectors and matrices to tensors
• Final remarks and conclusions
Semantics and Word Order

- It is estimated that the meaning of English comes from:
  - Word choice → 80%
  - Word order → 20%

Word Order in Additive Models

- Additive composition can be sensitive to word order by weighting the word contributions*

\[ p = x + y \]

\[ p = \alpha x + \beta y \]

Circular Convolution Model

- Word order encoded into a vector by collapsing outer-product matrix of word vectors

\[ p_i = \sum_j x_j \cdot y_{(i-j) \mod n} \]

\[ p_i = (p_0, p_1, p_2) \]

*Jones M.N. and Mewhort D.J.K (2007), Representing word meaning and order information in a composite holographic lexicon, Psychological Review, 114, pp. 1-37*
The Random Permutation Model

- Use permutation functions to randomly shuffle the vectors to be composed*

\[ p = Mx + M^2y \]

Recursive Matrix Vector Spaces

• Each word and phrase is represented by a vector and a matrix*

\[ p_1 = f_v(Zp_o, P_oZ) \]
\[ P_1 = f_M(P_o, Z) \] \((p_1, P_1)\)

\[ p_o = f_v(Yx, Xy) \]
\[ P_o = f_M(X, Y) \]

\((p_o, P_o)\)

\((x, X)\) \((y, Y)\) \((z, Z)\)

Section 4

Future Research and Applications

- Current limitations of vector space models
- Encoding word position information into vectors
- **From vectors and matrices to tensors**
- Final remarks and conclusions
Union/Intersection Limited Binding

- Multiplicative operations limit vector interaction to those common non-zero components only
  \[ [1, 0, 3, 0, 1, 0] \times [0, 2, 1, 0, 4, 0] = [0, 0, 3, 0, 4, 0] \]

- Additive operations limit vector interaction to both common and non-common non-zero components
  \[ [1, 0, 3, 0, 1, 0] + [0, 2, 1, 0, 4, 0] = [1, 2, 3, 0, 4, 0] \]

- Can we define operations to model richer interactions across vector components?
Vector Binding with Tensor Product*

- The tensor product of two vectors

\[ a \otimes b = \{ a_i b_j \} \text{ for } i = 1, 2 \ldots N_a \text{ and } j = 1, 2 \ldots N_b \]

- All possible interactions across components are taken into account

- But, the resulting vector representation is of higher dimensionality!

* Smolensky P. (1990), Tensor product variable binding and the representation of symbolic structures in connectionist systems, Artificial Intelligence, 46, pp.159-216
Compressing Tensor Products

- Compress the result to produce a composed representations with the same dimensionality of the original vector space

- One representative example of this is the circular convolution model

- *Can tensor representations be exploited at high dimensional space?*
Section 4

Future Research and Applications

- Current limitations of vector space models
- Encoding word position information into vectors
- From vectors and matrices to tensors
- Final remarks and conclusions
VSMs in Monolingual Applications

Vector Space Models have been proven useful for many monolingual NLP applications, such as:

- Clustering
- Classification
- Information Retrieval
- Question Answering
- Essay grading
- Spelling Correction
- Role Labeling
- Sense Disambiguation
- Information Extraction
- and so on...
VSMs in Cross-language Applications

Vector Space Models are also starting to be proven useful for cross-language NLP applications:

- Cross-language information retrieval
- Cross-script information retrieval
- Parallel corpus extraction and generation
- Automated bilingual dictionary generation
- Machine Translation (decoding and evaluation)
- Cross-language plagiarism detection
Future Research

Seems to be moving in two main directions:

• Improving the representation capability of current VSM approaches by:
  • Using neural network architectures
  • Incorporating word order information
  • Leveraging on more complex operators
• Developing a more comprehensive framework by combining formal and distributional approaches
Section 4

Main references for this section

- M. N. Jones and D. J. K. Mewhort, 2007, “Representing word meaning and order information in a composite holographic lexicon”
- M. Sahlgren, A. Holst and P. Kanerva, 2008, “Permutations as a means to encode order in word space”
Section 4

Additional references for this section


- Smolensky P. (1990), Tensor product variable binding and the representation of symbolic structures in connectionist systems, Artificial Intelligence, 46, pp.159-216