On metric embedding for boosting semantic similarity computations

J. Subercaze, C. Gravier, F. Laforest

Laboratoire Hubert Curien
Université Jean Monnet
Télécom Saint-Étienne, France

LABORATOIRE Hubert CURIEN
UMR • CNRS • 5516 • ST-ETIENNE

INSTITUT Mines-Télécom

Word to Word Semantic Similarity

Distributional semantics

Word Embedding: learn a real-valued vector representation of words so that any vector distance – usually the cosine similarity – encodes the word-to-word semantic similarity

Knowledge base semantic similarity

Uses a taxonomy, usually Wordnet to compute semantic similarity between words. Methods based on graph metrics or information content

- **Graph metrics** HSO [Hirst and St-Onge, 1998], LCH [Leacock and Chodorow, 1998], WUP [Wu and Palmer, 1994]
- **Information Content** LESK [Banerjee and Pedersen, 2002], JCN [Jiang and Conrath, 1997]
- **Hybrid** RES [Resnik, 1995], LIN [Lin, 1998]

Performance

Quality

JCN and LCH present the best correlation with human ranking

Runtime

Both methods are slow, tens/hundreds of milliseconds for cold start, milliseconds afterward.

Going Faster

Finding Binary Codes

Is it possible to find binary codes so that their hamming distance preserve the semantic similarity ?

Going Faster

Finding Binary Codes

Is it possible to find binary codes so that their hamming distance preserve the semantic similarity ?

Hint

For LCH, the similarity is a monotonic function of the shortest path distance in the Wordnet hypernym structure. => Metric Embedding

Metric Embedding

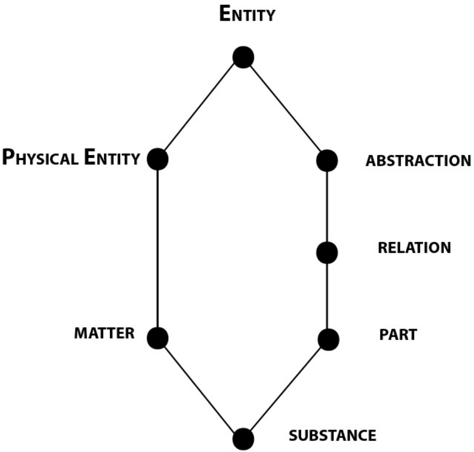
Definition

A metric embedding function f from a metric space (A, d_1) into (B, d_2) the is defined as follows

$$\forall(w_i, w_j), d_1(w_i, w_j) = \lambda \cdot d_2(f(w_i), f(w_j))$$

$w_i, w_j \in A$
λ is a scalar

Wordnet Hypernyms: a lattice



Sample lattice from Wordnet

Embedding, first try

Lattice into Hypercube

Deza and Laurent (1997) showed that a lattice with shortest path distance can be isometrically embedded into an hypercube of 2^n dimensions.

Issues

Dimensions too high: $\approx 2^{84.000}$ for Wordnet Synsets. Not a constructive proof.

Embedding, second try

Lattice is too complicated. What about a tree ?

Embedding, second try

Lattice is too complicated. What about a tree ?

We can obtain a tree from the poset by cutting 1% of the edges

Embedding, second try

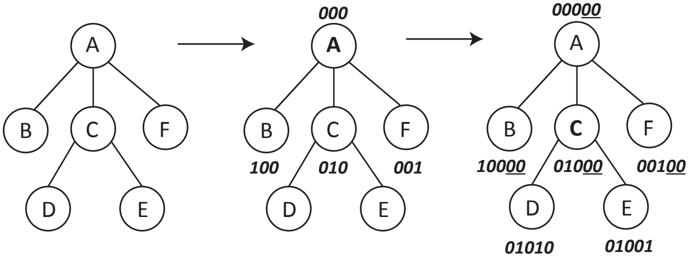
Lattice is too complicated. What about a tree ?

We can obtain a tree from the poset by cutting 1% of the edges

What the theory says

Isometric embedding: $n-1$ dimensions.

Isometric embedding of a tree



Construction of isometric embedding on a sample tree. For this six nodes tree, the embedding requires five bits.

Let's relax

Non isometric embedding

The question is to construct an embedding with a *good* distance preservation.
I.e. high correlation with original pairwise distances.

Let's relax

Non isometric embedding

The question is to construct an embedding with a *good* distance preservation.
I.e. high correlation with original pairwise distances.

Huge search space

$$C = \frac{(2^n)!}{(n-r)!}$$

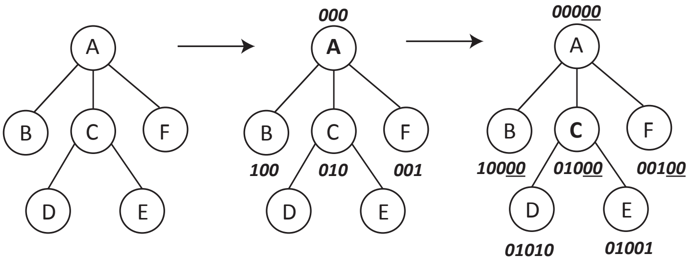
For 84K nodes (n) into a 64 (r) dimensional hypercube: $C > 10^{100,000}$

Let's be specific

Wordnet data

- *Branching factor* AVG: 4.9 - STD: 14. 96% of nodes < 20.
- *Depth* AVG: 8.5 - STD: 2. MAX: 18.

Let's recall



A
B
C
F
D
E
000
A
B
C
F
100
010
001
D
E
00000
A
B
C
F
10000
01000
00100
D
E
01010
01001

Heuristic

We choose to preserve the parent-child distance instead of siblings distance.

Heuristic

We choose to preserve the parent-child distance instead of siblings distance.

Unique signature

For each node with k children, we allocate $\lceil log_2(k+1) \rceil$ bits. Use best extension first (i.e respecting both distances).

Heuristic

We choose to preserve the parent-child distance instead of siblings distance.

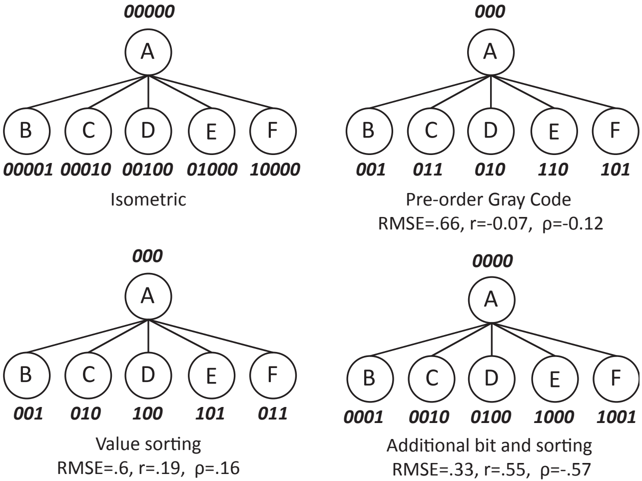
Unique signature

For each node with k children, we allocate $\lceil log_2(k+1) \rceil$ bits. Use best extension first (i.e respecting both distances).

Word alignment

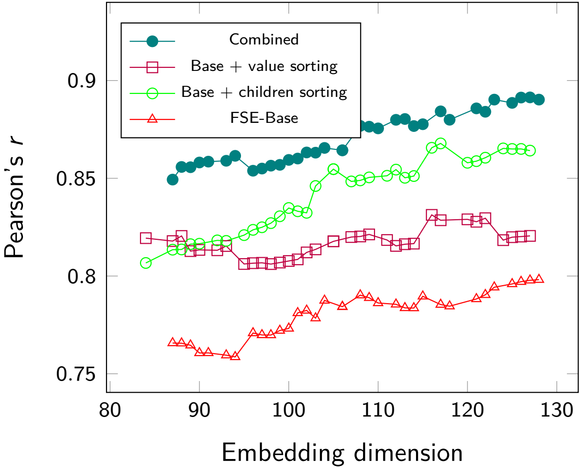
If the obtained embedding is not word aligned, we can use the remaining bits to enhance the embedding.

Example



Approaches to reduce the tree embedding dimensions.

Numerical Experiment I



FSE: influence of optimizations and dimensions on the correlation over the tree distance on Wordnet.

Numerical Experiment II

Embedding	Bits	Pearson's r	Spearman's ρ
Chen et al.	17	.235	.186
FSE-Base	84	.699	.707
FSE-Best	**128**	**.819**	**.829**
Isometric	84K	.919	.931

Correlations between LCH, isometric embedding, and FSE for all distances on all Wordnet-Core noun pairs (p-values $\leq 10^{-14}$).

Numerical Experiment III

Algo	Measure	Amount of pairs (n)				
		10^3	10^4	10^5	10^6	10^7
WS4J	$10^3 \cdot$ ms	0.156	1.196	11.32	123.89	1,129.3
FSE-Best	ms	0.04	0.59	14.15	150.58	1,482
speedup		$\times$3900	$\times$2027	$\times$800	$\times$822	$\times$762

Running time or pairwise similarity computations.

Application

Similar Sentence retrieval

Find semantic similar sentences using the hash values. Hash of a sentence is obtained using Simhash. [Bamba et al., 2012, Subercaze et al., 2013]

Application

Similar Sentence retrieval

Find semantic similar sentences using the hash values. Hash of a sentence is obtained using Simhash. [Bamba et al., 2012, Subercaze et al., 2013]

Example

Token	Weight	Hash
a	3	1 0 1 1 0 1
b	2	0 1 1 0 0 1
c	1	1 0 0 1 1 1

Application

Similar Sentence retrieval

Find semantic similar sentences using the hash values. Hash of a sentence is obtained using Simhash. [Bamba et al., 2012, Subercaze et al., 2013]

Example - Set bit value to +/- weight

Token	Weight	Hash
a	3	3 -3 3 3 -3 3
b	2	-2 2 2 -2 -2 2
c	1	1 -1 -1 1 1 1

Application

Similar Sentence retrieval

Find semantic similar sentences using the hash values. Hash of a sentence is obtained using Simhash. [Bamba et al., 2012, Subercaze et al., 2013]

Example - Sum the values

Token	Weight	Hash
a	3	3 -3 3 3 -3 3
b	2	-2 2 2 -2 -2 2
c	1	1 -1 -1 1 1 1
total		2 -2 4 2 -4 6

Application

Similar Sentence retrieval

Find semantic similar sentences using the hash values. Hash of a sentence is obtained using Simhash. [Bamba et al., 2012, Subercaze et al., 2013]

Example - Final hash

Token	Weight	Hash
a	3	3 -3 3 3 -3 3
b	2	-2 2 2 -2 -2 2
c	1	1 -1 -1 1 1 1
total		2 -2 4 2 -4 6
hash		1 0 1 1 0 1

Demonstration

Semantic similarity: short sentences.

Bamba, P., Subercaze, J., Gravier, C., Benmira, N., and Fontaine, J. (2012).
The twitaholic next door.: scalable friend recommender system using a concept-sensitive hash function.
In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 2275–2278.

Banerjee, S. and Pedersen, T. (2002).
An adapted lesk algorithm for word sense disambiguation using wordnet.
In *Computational linguistics and intelligent text processing*, pages 136–145. Springer.

Hirst, G. and St-Onge, D. (1998).
Lexical chains as representations of context for the detection and correction of malapropisms.
WordNet: An electronic lexical database, 305:305–332.

Jiang, J. J. and Conrath, D. W. (1997).
Semantic similarity based on corpus statistics and lexical taxonomy.
Proceedings of the 10th Research on Computational Linguistics International Conference.

Leacock, C. and Chodorow, M. (1998).
Combining local context and wordnet similarity for word sense identification.
WordNet: An electronic lexical database, 49(2):265–283.

Lin, D. (1998).
An information-theoretic definition of similarity.
In *ICML*, volume 98, pages 296–304.

Resnik, P. (1995).
Using information content to evaluate semantic similarity in a taxonomy.
In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Subercaze, J., Gravier, C., and Laforest, F. (2013).
Towards an expressive and scalable twitter's users profiles.
In *Web Intelligence*, pages 101–108.

Wu, Z. and Palmer, M. (1994).
Verbs semantics and lexical selection.
In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.