THE OHIO STATE UNIVERSITY | Department of Linguistics

AZMAT: Sentence Similarity Using Associative Matrices

Evan Jaffe, Lifeng Jin, David King, Marten Van Schijndel

Overview

- SemEval-2015 Task 2: English STS
  - Model human judgements of sentential similarity (0-5)
- SVM with linear kernel
  - **U**nfolding **R**ecursive **A**uto**E**ncoder (Socher et al., 2011)
  - **A**ssociative **M**atrices (Anderson et al., 1977; Howard and Kahana, 2002)
  - **G**loVe global vectors (Pennington et al., 2014)
  - **S**urface lexical overlap
- Training input is previous SemEval tasks 2012-2014
- Ranks 69th out of 74 systems
- Question: Does phrasal cosine similarity help?

Subsystem Combination

- Generate an embedding for every node of a binarized phrase-structure tree
- Measure cosine similarity of every node in sentence A to every node in sentence B
- Generate a fixed-length feature vector that concatenates ordered similarity scores, repeating out to required space, sorting

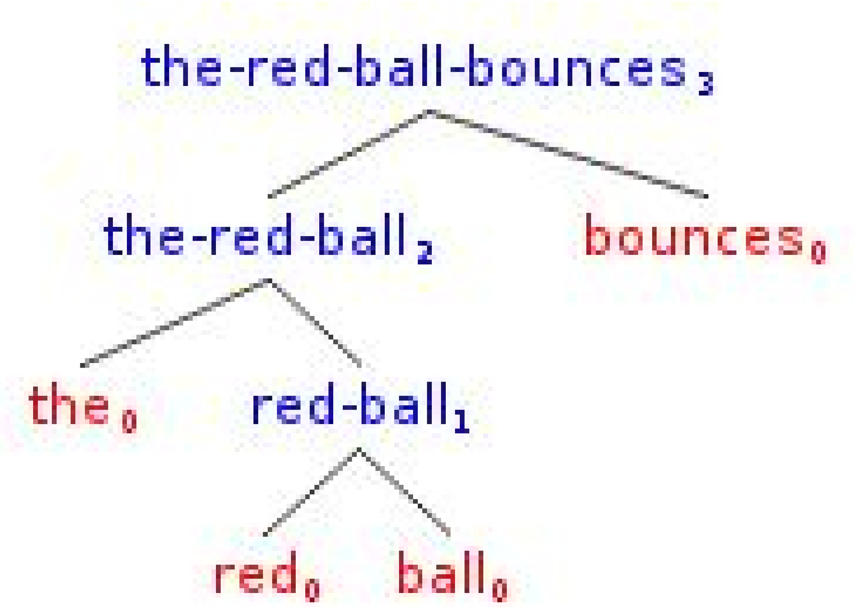

Fig 1: Binarized phrase-structure parse tree. Subscript denotes depth, where leaves are depth 0. Parents are the depth of their deepest child + 1.

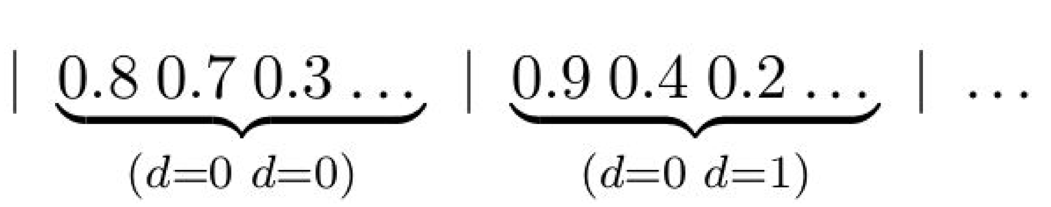

Fig 2: Fixed-length feature vector. Within the feature vector are subvectors that all have the same depth combination. For example, all the leaf-to-leaf node similarities would be ordered by similarity and placed in the depth 0, depth 0 section.

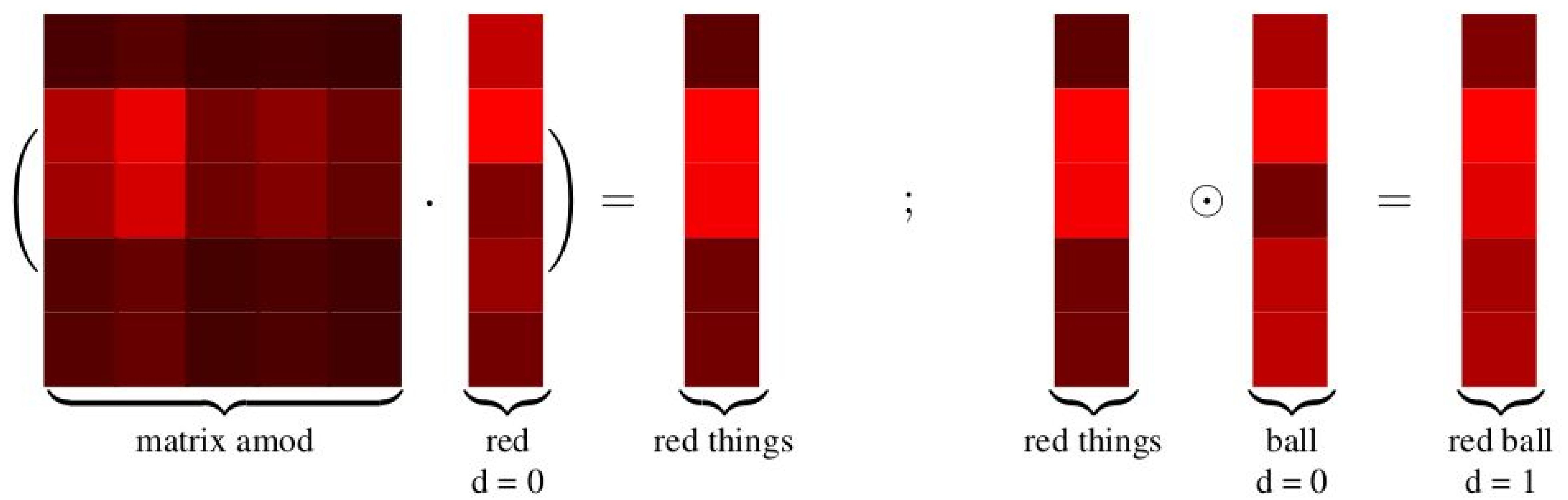

Fig 3: Step 1 of composition - embedding for 'red' leaf node multiplied by 'amod' relation matrix to generate intermediate 'red things' embedding. Step 2 of composition - intermediate 'red things' embedding pointwise multiplied by 'ball' leaf embedding to generate final phrasal embedding, 'red ball'.

Associative Matrices

- Used in memory and sentence processing models
- Uses binarized Generalized Categorial Grammar (GCG) parse tree
- Learns a matrix for each dependency type
- Generates a phrasal embedding for any given dependency triple, e.g., *<red, ball, amod>*

$$M_{deplabel} = \sum_{d \in D} (\bar{u}_d \otimes \bar{v}_d)$$

Eq 1: "Training" matrices. For each labeled dependency triple d in data D, sum the outer product of the vectors for the head u_d and dependent v_d into the appropriate matrix for the dependency label. E.g., for a labeled dependency triple <green, ball, amod> seen in training, add the outer product of the 'green' and 'ball' vectors to the 'amod' matrix.

Unfolding Recursive Autoencoders

- Used in paraphrase detection
- Composes embeddings for each node in a binary phrase-structure tree, given leaf embeddings
- Learns to encode and decode, with objective of minimizing reconstruction error
- Uses Stanford parser, not GCG tree
- Current work replaces dynamic pooling with depth-sensitive vector expansion in order to avoid lossy operations while retaining global structural similarity

GloVe Global Vectors

- 300-dimensional vectors trained on 42 billion tokens
- Composition just percolates up the head word from a binarized Generalized Categorial Grammar (GCG) parse tree

Surface Features

- SVM input features that do not use parse tree, do not use embeddings
- 1-3 exact/stem precision/recall lexical overlap

STS Experiment

- 3000 (sampled from 8500) sentence pairs from 5 domains
- 1000 pair cross-validation for post-hoc dev analysis

Model	Dev Unknown Domain ρ	Dev Known Domain ρ	STS Test ρ
SUGA	.537	.611	**.451**
UGA	.462	.549	-
SUA	.555	.623	-
SGA	.565	.630	-
SUG	.590	.657	-

Table 1: Mean Pearson correlation by model. S is surface features, U is unfolding recursive autoencoders, G is GloVe, and A is associative matrices.

	Leaf	Comp	Cross	Full
Mean	**.591**	.530	.510	.424
Wt. Mean	**.610**	.550	.521	.450

Table 2: Pearson correlation by cosine similarity type. Leaf only includes leaf-to-leaf similarities. Comp means phrasal-to-phrasal similarities only. Cross means leaf-phrasal similarities. Full includes all similarity features. Mean is across all 5 domains, Wt. Mean is the weighted mean by how many questions come from each domain.

Conclusions

- Leaf features are better predictors than phrasal comparison features
- Overfitting evident from development analysis
- Surface features are complementary to cosine similarity features (SUGA vs. UGA model performance)
- Composition with associative matrices does not seem to work well

Discussion

- Possible matrix saturation with too few dependency labels
- Finer-grained syntactic info (beyond depth) when grouping cosine similarities
- SVM regularization tuning
- Using phrasal nodes to do similarity is an open challenge

Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1343012. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would also like to thank the anonymous reviewers for their helpful suggestions and comments.

Contact Information

If you have any questions/comments/concerns, you may reach us at the following:

Evan Jaffe: jaffe.59@osu.edu

Lifeng Jin: jin.544@osu.edu

David King: king.2138@osu.edu

Marten Van Schijndel: vanschm@ling.osu.edu