Module 10.9: Evaluating word representations

How do we evaluate the learned word representations ?

Semantic Relatedness

Semantic Relatedness

- Ask humans to judge the relatedness between a pair of words

$S_{human}(cat, dog) = 0.8$

Semantic Relatedness

- Ask humans to judge the relatedness between a pair of words
- Compute the cosine similarity between the corresponding word vectors learned by the model

$$S_{human}(cat, dog) = 0.8$$

$$S_{model}(cat, dog) = \frac{v_{cat}^T v_{dog}}{\| v_{cat} \| \| v_{dog} \|} = 0.7$$

Semantic Relatedness

- Ask humans to judge the relatedness between a pair of words
- Compute the cosine similarity between the corresponding word vectors learned by the model
- Given a large number of such word pairs, compute the correlation between S_{model} & S_{human}, and compare different models

$$S_{human}(cat, dog) = 0.8$$

$$S_{model}(cat, dog) = \frac{v_{cat}^T v_{dog}}{\| v_{cat} \| \| v_{dog} \|} = 0.7$$

$$S_{human}(cat, dog) = 0.8$$

$$S_{model}(cat, dog) = \frac{v_{cat}^T v_{dog}}{\| v_{cat} \| \| v_{dog} \|} = 0.7$$

Semantic Relatedness

- Ask humans to judge the relatedness between a pair of words
- Compute the cosine similarity between the corresponding word vectors learned by the model
- Given a large number of such word pairs, compute the correlation between S_{model} & S_{human}, and compare different models
- Model 1 is better than Model 2 if

$$correlation(S_{model1}, S_{human}) > correlation(S_{model2}, S_{human})$$

Synonym Detection

Synonym Detection

- Given: a term and four candidate synonyms

Term : levied

Candidates : {unposed,
believed, requested, correlated}

Synonym Detection

- Given: a term and four candidate synonyms
- Pick the candidate which has the largest cosine similarity with the term

Term : levied

Candidates : {unposed, believed, requested, correlated}

$$\textbf{Synonym} := \underset{c \in C}{argmax}\ cosine(v_{term}, v_c)$$

Term : levied

Candidates : {unposed,
believed, requested, correlated}

$$\textbf{Synonym} := \underset{c \in C}{argmax}\ cosine(v_{term}, v_c)$$

Synonym Detection

- Given: a term and four candidate synonyms
- Pick the candidate which has the largest cosine similarity with the term
- Compute the accuracy of different models and compare

Analogy

Analogy

- Semantic Analogy: Find nearest neighbour of $v_{brother} - v_{sister} + v_{grandson}$

brother : sister :: grandson : ?

Analogy

- Semantic Analogy: Find nearest neighbour of $v_{brother} - v_{sister} + v_{grandson}$
- Syntactic Analogy: Find nearest neighbour of $V_{work} - v_{works} + v_{speak}$

brother : sister :: grandson : ?
work : works :: speak : ?

- So which algorithm gives the best result ?

- So which algorithm gives the best result ?
- Boroni et.al [2014] showed that predict models consistently outperform count models in all tasks.

- So which algorithm gives the best result ?
- Boroni et.al [2014] showed that predict models consistently outperform count models in all tasks.
- Levy et.al [2015] do a much more through analysis (IMO) and show that good old SVD does better than prediction based models on similarity tasks but not on analogy tasks.