New Word Pair Level Embeddings to Improve Word Pair Similarity

Asma Shaukat, *Nazar Khan*

{asma.shaukat,nazarkhan}@pucit.edu.pk

http://faculty.pucit.edu.pk/nazarkhan/

Computer Vision & Machine Learning Group
Punjab University College of Information Technology (PUCIT)
Lahore, Pakistan

ICDAR Workshop on Machine Learning (ICDAR-WML 2017)
Kyoto, Japan
11th November 2017

Word Pair Similarity

- The problem of quantifying the similarity level between two words is known as the *Word Pair Similarity (WPS)* problem.

Word1	**Word2**	**Similarity score [0,50]**
car	automobile	50
cat	kitten	49
bakery	zebra	0
evening	walk	27
apartment	valley	14

Table: Some manually marked similarity estimates from the MEN-3000 dataset.

Word Pair Similarity

WPS plays an important role in natural language processing and information retrieval such as

- Sentence pair similarity
- Document summarization
- Automatic thesauri generation
- Automatic retrieval of similar words
- Query expansion
- Automatic analogy solving

WPS for image/video similarity

As long as we are willing to "look" at images using text.

Two Types of Similarities Between Words

Medin *et al.* '90

1. **Attributional Similarity**: similar words share high degree of attributes.
   - Car is similar to a truck because they share some attributes (tires, driving, engine, etc.).
   - Lion is similar to a tiger because they share some attributes (big cats, claws, sharp teeth).
   - Cat is similar to a dog because both are pets.
2. **Relational Similarity**: words can be connected with each other through different relations.
   - Bright is the *antonym of* dark.
   - Smoking *causes* cancer.

Attributional vs Relational Similarity

Happy – Sad – Surprised

- Happy, sad and disgusted share a common *attribute* of being human expressions.
- However, happy and sad share an additional *relationship* of being antonyms.
- Therefore, similarity between happy and sad should be diferent from similarity between happy and surprised.
- We explicitly exploit such relational information for computation of similarity.

Basic Idea

Embed words as points in a (continuous) vector space. Learn embeddings so that similar words are represented by similar vectors.

Figure: Groups of words which share similar meaning should be embedded in similar vector space locations. From nlp.cs.tamu.edu/resources/wordvectors.ppt.

Power of Word Embeddings

Word2Vec (Mikolov *et al.*), GLOVE (Pennington *et al.*)

- They possess valuable information about words and their relations with each other.
- vector[Queen] - vector[King] = vector[Woman] - vector[Man]

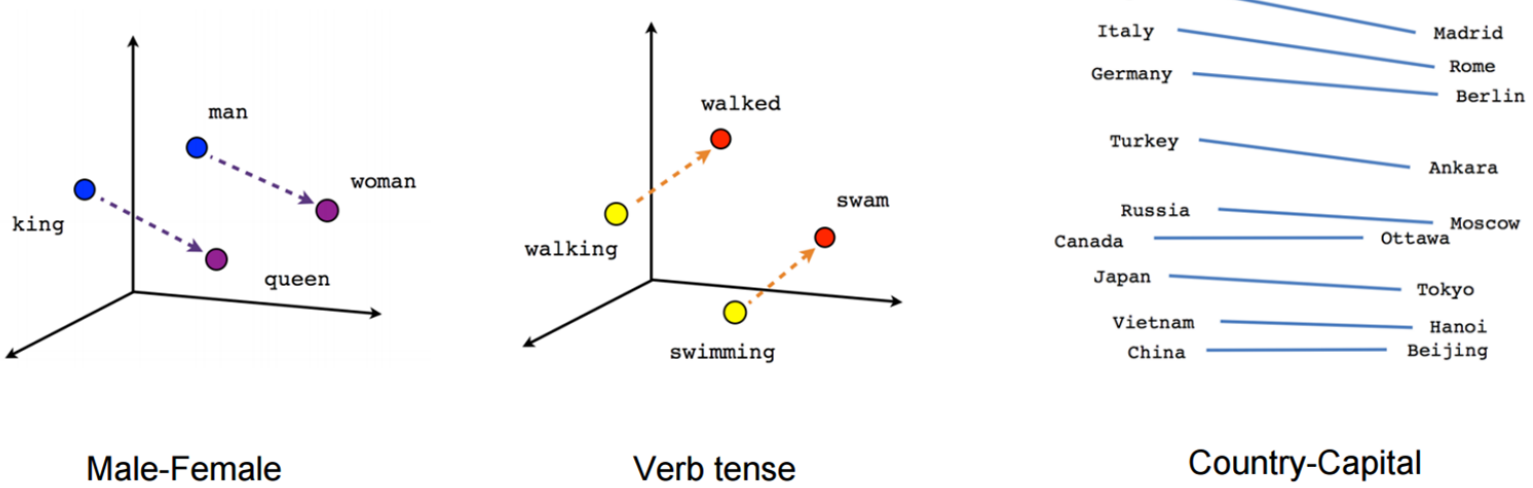


Figure: Embeddings should also reflect relationships between words. From www.tensorflow.org/tutorials/word2vec.

Well-known Word Embeddings

Learned from large text corpora

- Word2Vec
- GLOVE

Learned from structured semantic networks

- ConceptNet Numberbatch (CN16 and CN17)

Visualization of Word Embeddings

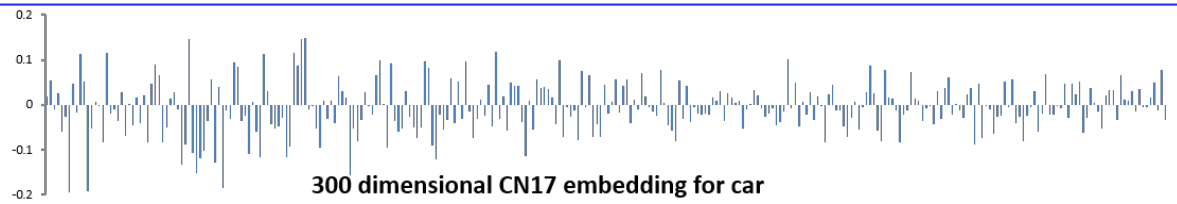
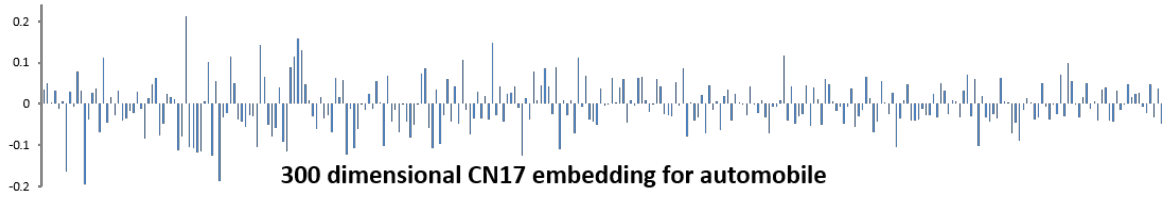
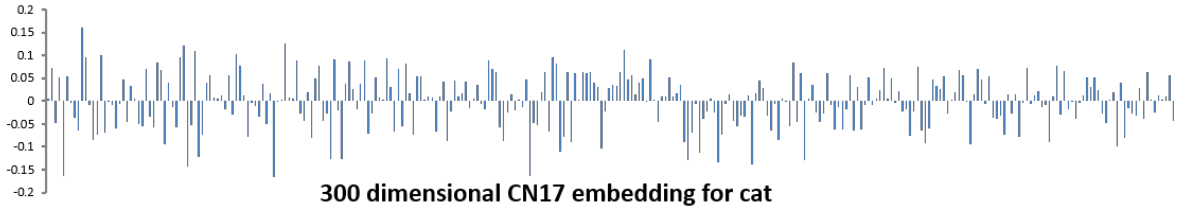
Car	
Automobile	
Cat	

Table: Examples of Conceptnet Numberbatch word embeddings for car, automobile and cat. Cosine similarity between car and automobile embeddings is 0.8933. In contrast, embeddings of car and cat have similarity of only 0.1257.

Current Approach

1. Embed individual words into a vector space.
2. WPS = cosine similarity of individual word vectors.

Observations

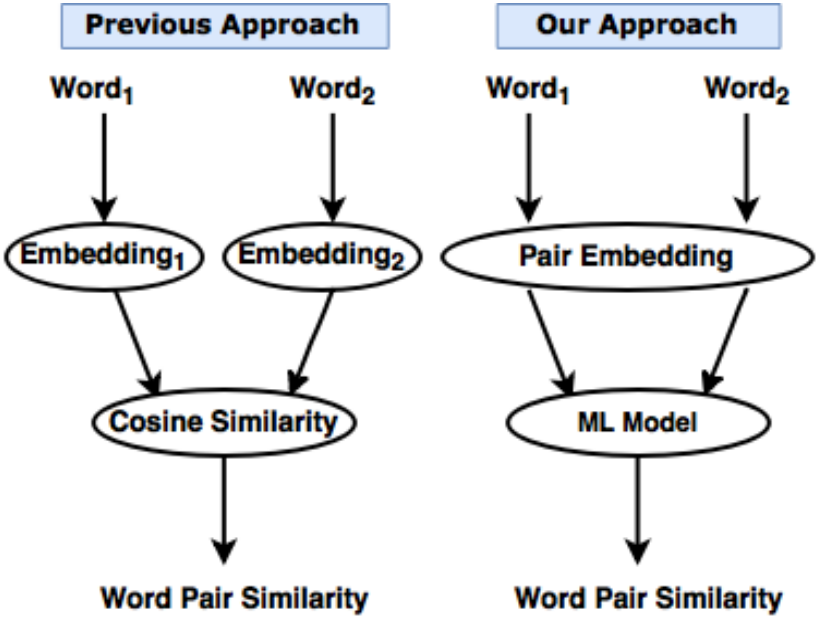
Why stop at current embeddings?
Why stop at cosine similarity?

Alternatives

1. $\mathbf{e}_1, \mathbf{e}_2 \longrightarrow$ ML $\longrightarrow$ similarity value.
2. $\mathbf{e}_{12} \longrightarrow$ ML $\longrightarrow$ similarity value.

Option 2 is advantageous because

- it can explicitly capture relational information between words, and
- size of embedding can be controlled (which implies that complexity of ML model can be controlled).

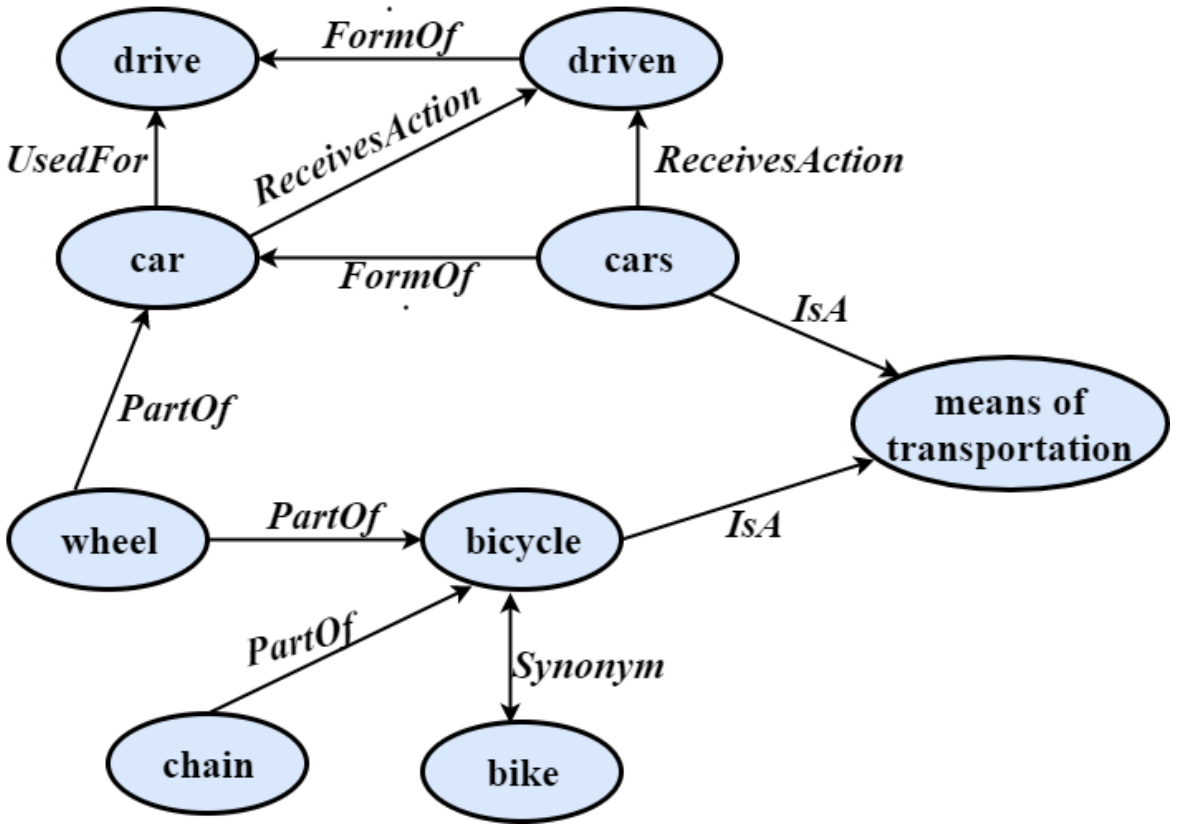


Word Pair Embeddings

For word pair (w_1, w_2), construct a 53 dimensional word pair embedding $\mathbf{v}$

- The first element of $\mathbf{v}$ is the dot-product of individual word embeddings.
- Last 52 elements of $\mathbf{v}$ are relationship features extracted from ConceptNet.

ConceptNet



A crowd-sourced semantic network of words and their relationships.
Adapted from https://blog.conceptnet.io/tag/conceptnet/.

ConceptNet

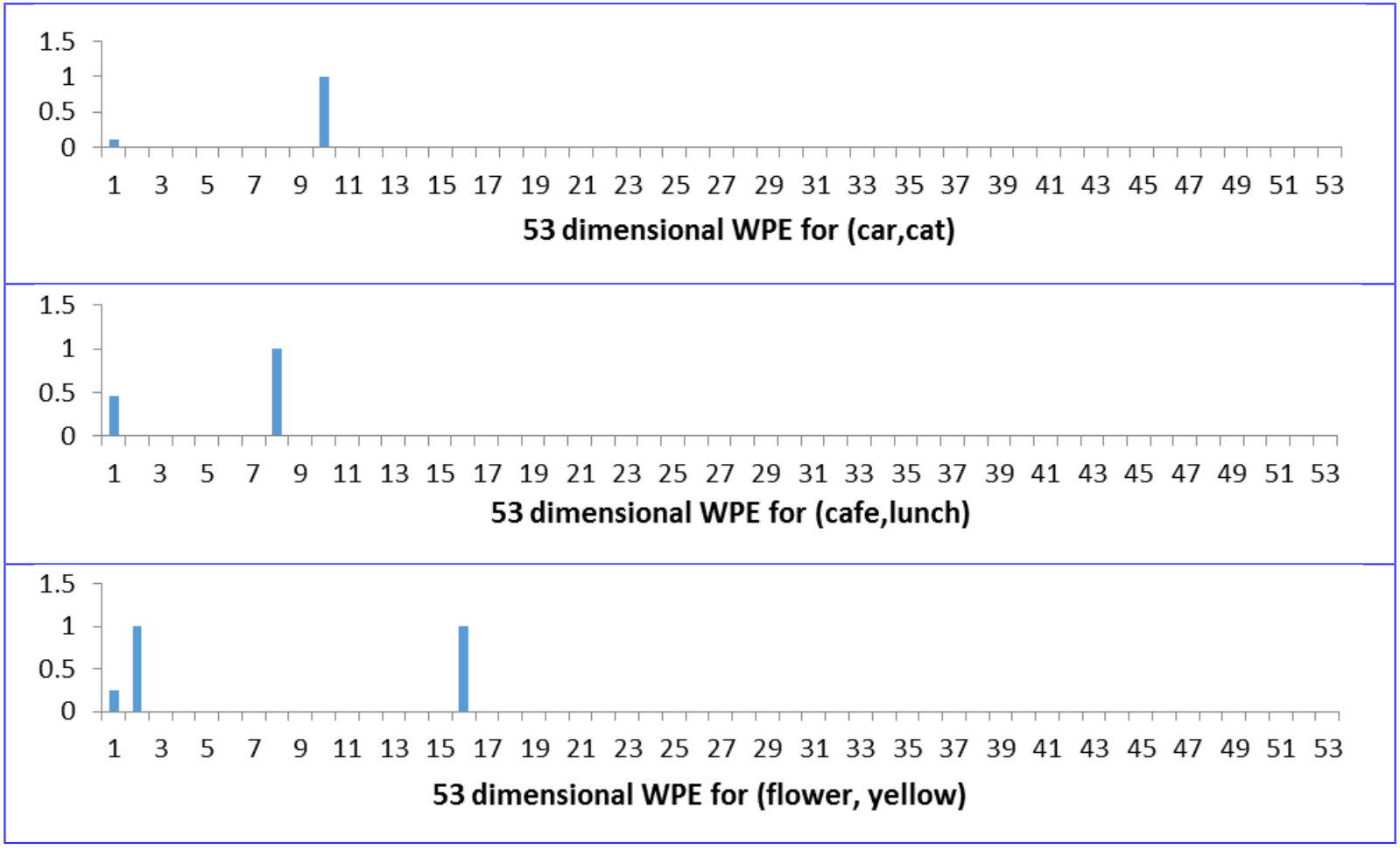
Common sense knowledge in ConceptNet for the word "car".

Word Pair Relations

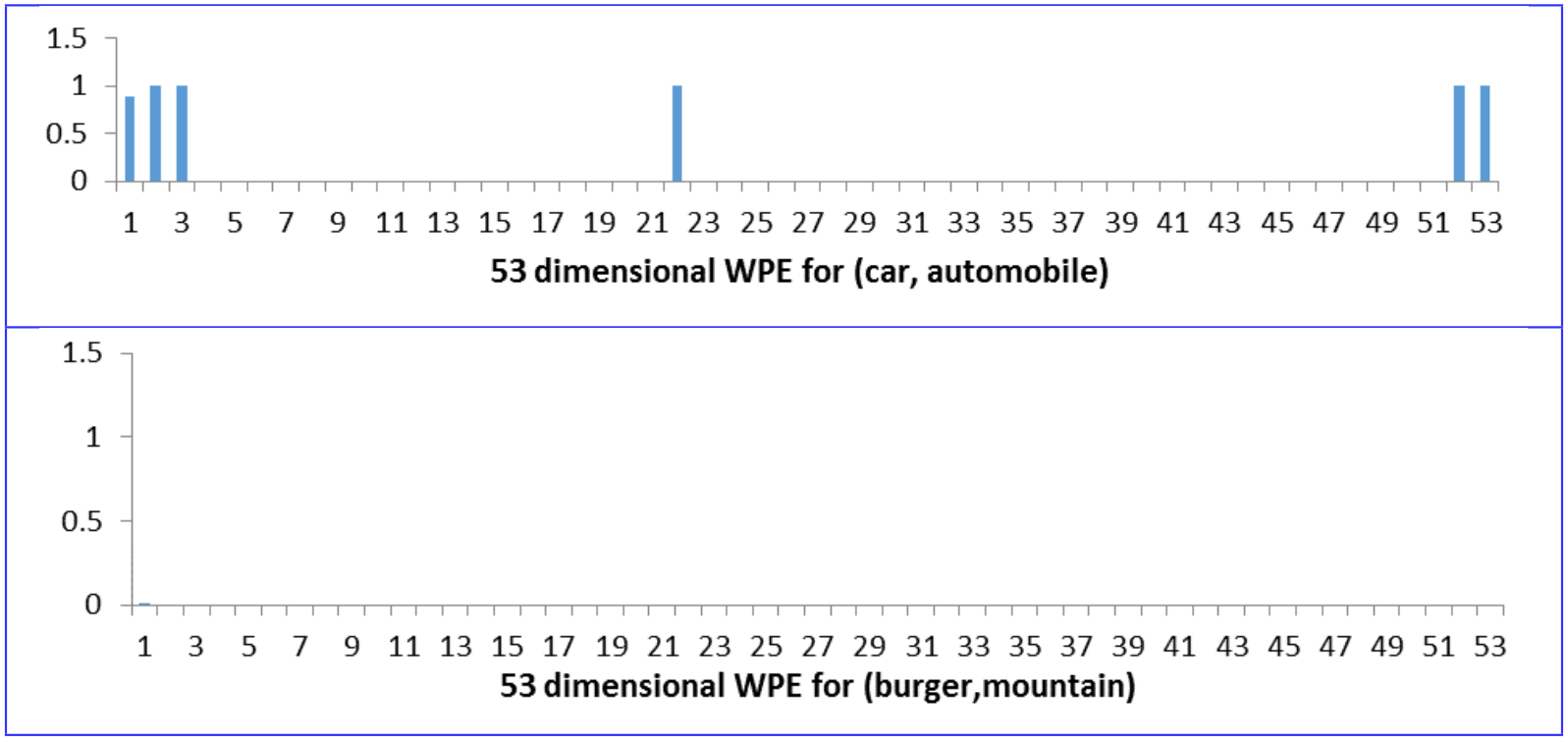
DefinedAs	HasProperty	SimilarTo	participleOf
RelatedTo	MotivatedByGoal	MadeOf	DistinctFrom
IsA	ObstructedBy	SymbolOf	EtymologicallyRelatedTo
PartOf	Desires	genre	FormOf
MemberOf	NotDesires	LocatedNear	NotHasProperty
HasA	CreatedBy	HasContext	NotCapableOf
UsedFor	Synonym	MemberOf	MannerOf
CapableOf	Antonym	Attribute	NotUsedFor
AtLocation	DerivedFrom	adverbPertainsTo	ExternalURL
Causes	TranslationOf	CompoundDerivedFrom	TwoPhraseRelation
HasSubevent	ReceivesAction	CausesDesire	
HasFirstSubevent	EtymologicallyDerivedFrom	DesireOf	
HasLastSubevent	InstanceOf	adjectivePertainsTo	
HasPrerequisite	Entails	knownFor	

Table: List of 52 features used from ConceptNet to capture relations between word pairs.

Visualization of Word Pair Embeddings



Visualization of Word Pair Embeddings



ML Models

- Our word pair embeddings can be used as inputs to any regression based machine learning model.
- Human marked similarity values can be used as targets for training.
- We use 2 simple models:
  1. Regularized Linear Regression
     - Cross-validated the regularization hyperparameter in the range 0.001 to 1000.
  2. Neural network with just 1 hidden layer containing only 3 neurons.
     - Bayesian regularization.

Datasets

1. MEN-3000
   - A gold standard dataset with 3000 word pairs.
   - All pairs with their crowd sourced similarity judgments.
   - Comes with development (2000) and testing (1000) splits.
   - Used to evaluate many WPS algorithms.
2. WordSim-353
   - 353 word pairs with their similarity judgments.
   - Dataset does not come with development and testing splits.
   - Similar to the MEN dataset, we picked every third word pair for testing.

Results
MEN Dataset

Embeddings	Model	Correlation
CN17	Cosine Similarity	0.8680
WPE(CN17)	Linear Regression	**0.8794**
WPE(CN17)	Neural Network	**0.8830**
CN16	Cosine Similarity	0.8612
WPE(CN16)	Linear Regression	**0.8736**
WPE(CN16)	Neural Network	**0.8794**
Glove42	Cosine Similarity	0.8143
WPE(Glove42)	Linear Regression	0.8109
WPE(Glove42)	Neural Network	**0.8538**

Table: Correlation between WPS predictions and ground-truth similarity estimates for MEN dataset using different embeddings and different models. CN17, CN16 and Glove42 are three different embeddings of individual words. WPE(X) denotes that embeddings from X were used to compute the cosine similarity feature only.

Results

WordSim-353 Dataset

Embeddings	Model	Correlation
CN17	Cosine Similarity	0.8385
WPE(CN17)	Linear Regression	**0.8524**
WPE(CN17)	Neural Network	**0.8435**
CN16	Cosine Similarity	0.8610
WPE(CN16)	Linear Regression	**0.8722**
WPE(CN16)	Neural Network	**0.8720**
Glove42	Cosine Similarity	0.8200
WPE(Glove42)	Linear Regression	0.7986
WPE(Glove42)	Neural Network	**0.8649**

Table: Correlation between WPS predictions and ground-truth similarity estimates for WordSim-353 dataset using different embeddings and different models. CN17, CN16 and Glove42 are three different embeddings of individual words. WPE(X) denotes that embeddings from X were used to compute the cosine similarity feature only.

Conclusion

- State-of-the-art embeddings of individual words mainly capture attributional similarity.
- Pairs of words have relational similarity.
- Embeddings used for WPS should be computed at the level of word pairs.
- Results on the MEN and WordSim-353 datasets show that word pair embeddings are better than individual word embeddings for WPS computation.

References

ConceptNet

- http://conceptnet.io/

Word embeddings

- https://code.google.com/archive/p/word2vec/
- https://nlp.stanford.edu/projects/glove/
- https://github.com/commonsense/conceptnet-numberbatch

Questions?