Word Embeddings

Deep Learning for NLP

Kevin Patel

ICON 2017

December 21, 2017

Outline

1. Introduction
2. Word Embeddings
   - Count Based Embeddings
   - Prediction Based Embeddings
   - Multilingual Word Embeddings
   - Interpretable Word Embeddings
3. Evaluating Word Embeddings
   - Intrinsic Evaluation
   - Extrinsic Evaluation
   - Evaluation Frameworks
   - Visualizing Word Embeddings

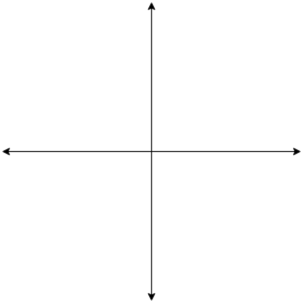
Outline

Layman(ish) Intro to ML

- In simple terms, Machine Learning comprises of
  - Representing data in some numeric form
  - Learning some function on that representation

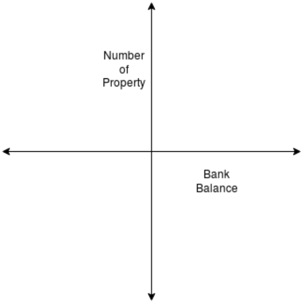
Layman(ish) Intro to ML

- In simple terms, Machine Learning comprises of
  - Representing data in some numeric form
  - Learning some function on that representation



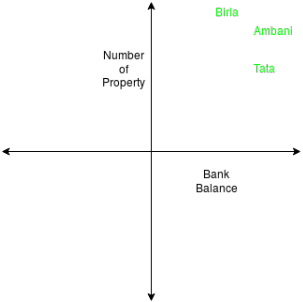
Layman(ish) Intro to ML

- In simple terms, Machine Learning comprises of
  - Representing data in some numeric form
  - Learning some function on that representation



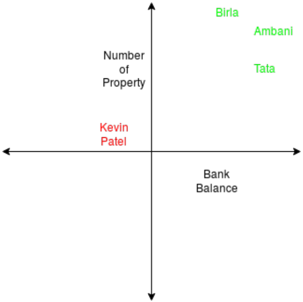
Layman(ish) Intro to ML

- In simple terms, Machine Learning comprises of
  - Representing data in some numeric form
  - Learning some function on that representation



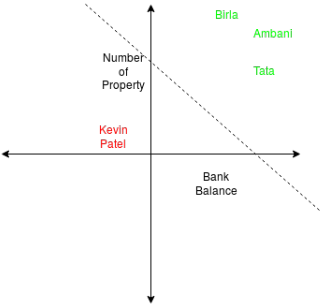
Layman(ish) Intro to ML

- In simple terms, Machine Learning comprises of
  - Representing data in some numeric form
  - Learning some function on that representation



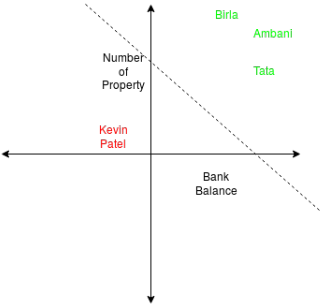
Layman(ish) Intro to ML

- In simple terms, Machine Learning comprises of
  - Representing data in some numeric form
  - Learning some function on that representation



Layman(ish) Intro to ML

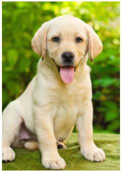
- In simple terms, Machine Learning comprises of
  - Representing data in some numeric form
  - Learning some function on that representation



- How to place words to learn, say, Binary Sentiment Classification?
  - Good: Positive
  - Awesome: Positive
  - Bad: Negative

Representations for Learning Algorithms

- Detect whether the following image is dog or not?



  - Basic idea: feed raw pixels as input vector
  - Works well:
    - Inherent structure in the image

- Detect whether a word is a dog or not?

Labrador

  - Nothing in spelling of *labrador* that can connect it to *dog*
  - Need a representation of *labrador* which indicates that it is a dog

Local Representations

- Information about a particular item located solely in the corresponding representational element (dimension)
- Effectively one unit is turned on in a network, all the others are off
- No sharing between represented data
- Each feature is independent
- No generalization on the basis of similarity between features

Distributed Representations

- Information about a particular item distributed among a set of (not necessarily) mutually exclusive representational elements (dimensions)
  - One item spread over multiple dimensions
  - One dimension contributing to multiple items
- A new input is processed similar to samples in training data which were similar
  - Better generalization

Distributed Representations: Example

Number	Local Representation	Distributed Representation
0	1 0 0 0 0 0 0 0	0 0 0
1	0 1 0 0 0 0 0 0	0 0 1
2	0 0 1 0 0 0 0 0	0 1 0
3	0 0 0 1 0 0 0 0	0 1 1
4	0 0 0 0 1 0 0 0	1 0 0
5	0 0 0 0 0 1 0 0	1 0 1
6	0 0 0 0 0 0 1 0	1 1 0
7	0 0 0 0 0 0 0 1	1 1 1

Word Embeddings : Intuition

- Word Embeddings: distributed vector representations of words such that the similarity among vectors correlate with semantic similarity among the corresponding words

  Given that sim(*dog*, *cat*) is more than sim(*dog*, *furniture*), $\cos(\overrightarrow{dog}, \overrightarrow{cat})$ is greater than $\cos(\overrightarrow{dog}, \overrightarrow{furniture})$

- Such similarity information uncovered from context

Word Embeddings : Intuition

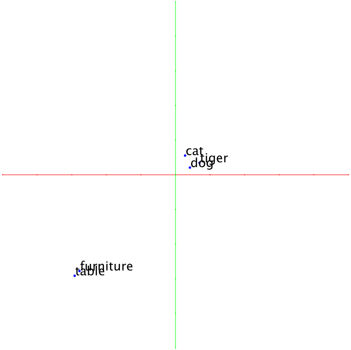
- Word Embeddings: distributed vector representations of words such that the similarity among vectors correlate with semantic similarity among the corresponding words

  Given that sim(*dog*, *cat*) is more than sim(*dog*, *furniture*), $\cos(\overrightarrow{dog}, \overrightarrow{cat})$ is greater than $\cos(\overrightarrow{dog}, \overrightarrow{furniture})$

- Such similarity information uncovered from context
- Consider the following sentences:
  - I like sweet food .
  - You like spicy food .
  - They like *xyzabc* food .
- What is *xyzabc* ?
- Meaning of words can be inferred from their neighbors (context) and words that share neighbors
  - Neighbors of *xyzabc*: {*like, food*}
  - Words that share neighbors of *xyzabc*: {*sweet, spicy*}

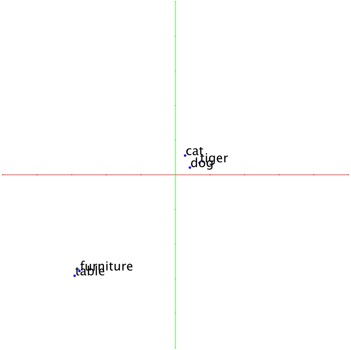
Modelling Meaning via Word Embeddings

- Geometric metaphor of meaning (Sahlgren, 2006):
  - Meanings are locations in semantic space, and semantic similarity is proximity between the locations.



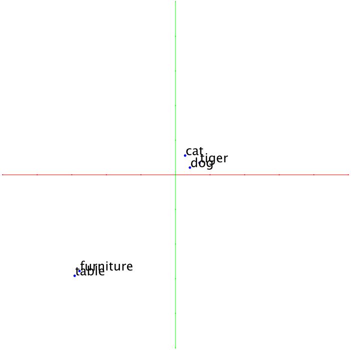
Modelling Meaning via Word Embeddings

- Geometric metaphor of meaning (Sahlgren, 2006):
  - Meanings are locations in semantic space, and semantic similarity is proximity between the locations.
- Distributional Hypothesis (Harris, 1970)
  - Words with similar distributional properties have similar meanings



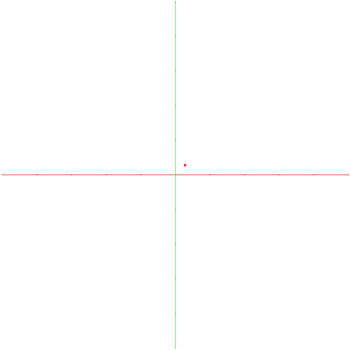
Modelling Meaning via Word Embeddings

- Geometric metaphor of meaning (Sahlgren, 2006):
  - Meanings are locations in semantic space, and semantic similarity is proximity between the locations.
- Distributional Hypothesis (Harris, 1970)
  - Words with similar distributional properties have similar meanings
  - **Only differences in meaning can be modelled**



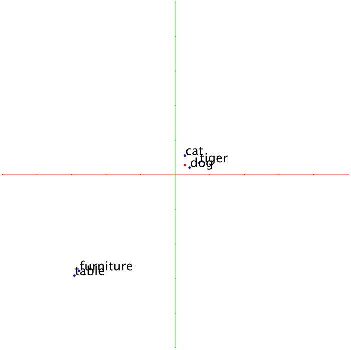
Modelling Meaning via Word Embeddings

- Geometric metaphor of meaning (Sahlgren, 2006):
  - Meanings are locations in semantic space, and semantic similarity is proximity between the locations.
- Distributional Hypothesis (Harris, 1970)
  - Words with similar distributional properties have similar meanings
  - **Only differences in meaning can be modelled**



Modelling Meaning via Word Embeddings

- Geometric metaphor of meaning (Sahlgren, 2006):
    - Meanings are locations in semantic space, and semantic similarity is proximity between the locations.
- Distributional Hypothesis (Harris, 1970)
    - Words with similar distributional properties have similar meanings
    - **Only differences in meaning can be modelled**



Entire Vector vs. Individual dimensions

- Only proximity in the entire space is represented
- No phenomenological correlations with dimensions of high-dimensional space (in majority of algorithms)
  - Those models who do have some correlations, are known as *interpretable models*

Modelling Meaning via Word Embeddings

- Co-occurrence matrix (Rubenstein and Goodenough, 1965)
  - A mechanism to capture distributional properties
  - Rows of co-occurrence matrix can be directly considered as word vectors
- Neural Word Embeddings
  - Vector representations learnt using neural networks - Bengio et al. (2003); Collobert and Weston (2008a); Mikolov et al. (2013b)

Co-occurrence Matrix

- Originally proposed by Schütze (1992)
- Foundation of count based approaches that follow
- Automatic derivation of vectors
- Collect co-occurrence counts in a matrix
- Rows or columns are the vectors of corresponding word
- If counting in both directions, matrix is symmetrical
- If counting in one side, matrix is asymmetrical, and is known as directional co-occurrence

Co-occurrence Matrix (contd.)

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

Co-occurrence Matrix

word	<>	I	like	love	hate	rate	rats	cats	dogs	bats
<>	0	4	0	0	0	0	1	1	1	1
I	4	0	1	1	1	1	0	0	0	0
like	0	1	0	0	0	0	0	1	0	0
love	0	1	0	0	0	0	0	0	1	0
hate	0	1	0	0	0	0	1	0	0	0
rate	0	1	0	0	0	0	0	0	0	1
rats	1	0	0	0	1	0	0	0	0	0
cats	1	0	1	0	0	0	0	0	0	0
dogs	1	0	0	1	0	0	0	0	0	0
bats	1	0	0	0	0	1	0	0	0	0

Word Embeddings
Count Based Embeddings

LSA

- Latent Semantic Analysis
- Originally developed as Latent Semantic Indexing (LSI) (Dumais et al., 1988)
- Adapted for word-space models
- Developed to tackle inability of models of co-occurrence matrices to handle synonymy
  - Query about *hotels* cannot retrieve results about *motels*
- Words and Documents dimensions $\rightarrow$ Latent dimensions
  - Uses Singular Value Decomposition (SVD) for dimensionality reduction

LSA (contd.)

- Words-by-documents matrix
- Entropy based weighting of co-occurrences

$$f_{ij} = (log(TF_{ij}) + 1) \times (1 - (\sum_j (\frac{p_{ij} log p_{ij}}{log D}))) \qquad (1)$$

  where D is number of documents, TF_{ij} is frequency of term i in document j, f_i is frequency of term i in document collection, and $p_{ij} = \frac{TF_{ij}}{f_i}$
- Truncated SVD to reduce dimensionality
- Cosine measure to compute vector similarities

HAL

- Hyperspace Analogous to Language (Lund and Burgess, 1996a)
- Developed specifically for word representations
- Uses directional co-occurrence

HAL (contd.)

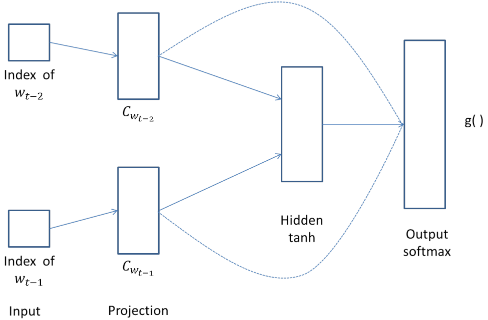
- Directional word by word matrix
- Distance weighting of the co-occurrences
- Concatenation of row-column vectors
- Dimensionality reduction optional
  - Discard low variant dimensions
- Normalization of vectors to unit length
- Similarities computed through either Manhattan or Euclidean distance

Word Embeddings
Prediction Based Embeddings

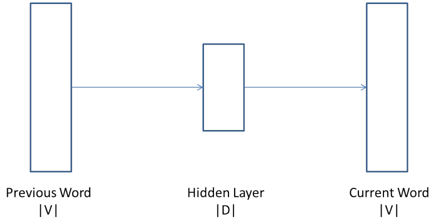
NNLM

- Neural Network Language Model
- Proposed by Bengio et al. (2003)
- Predict word given context
- Word Vectors learnt as a by-product of language modelling

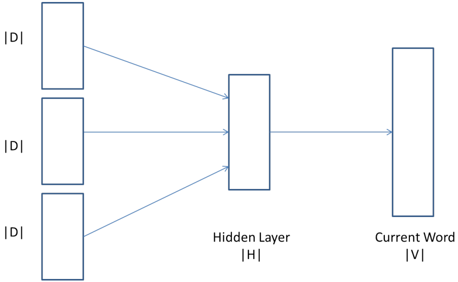
NNLM: Original Model



NNLM: Simplified (1)

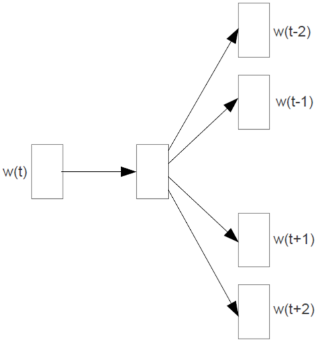


NNLM: Simplified (2)



Skip Gram

- Proposed by Mikolov et al. (2013b)
- Predict Context given word



Skip Gram (contd.)

- Given a sequence of training words $w_1, w_2, \ldots, w_T$, maximize

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j} \log p\left(w_{t+j}|w_t\right) \tag{2}$$

- where

$$p(w_O|w_I) = \frac{\exp(u_{w_O}^T v_{w_I})}{\sum_{w=1}^{W}\exp(u_w^T v_{w_I})} \tag{3}$$

Global Vectors (GloVe)

- Proposed by Pennington et al. (2014)
- Predict Context given word
- Similar to Skip-gram, but objective function is different

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \tag{4}$$

- where X_{ij} can be likelihood of i_{th} and j^{th} word occuring together, and f is a weightage function

Tuning word embeddings

- Techniques which intend to tune already trained word embeddings to various tasks using additional information
- Ling et al. (2015) improve quality of word2vec for syntactic tasks such as POS
  - Take word positioning into account
  - Structured Skip-Gram and Continuous Windows: available as wang2vec
- Levy and Goldberg (2014) use dependency parse trees
  - Linear windows capture broad topical similarities, and dependency context captures functional similarities
- Patel et al. (2017) use medical code hierarchy to improve medical domain specific word embeddings

Word Embeddings
Multilingual Word Embeddings

Objective

- English data $>>$ Data for other languages
- Language independent phenomenon learnt on English should be applicable to other languages
- Solution via word embeddings:
  - Project words of different languages into a common subspace
- Goal of multilingual word embeddings: Shared subspace for all languages
- Neural MT learns such embeddings implicitly by optimizing the MT objective
- We shall discuss explicit models
  - These models are for MT what word2vec, GloVe are for NLP
  - Much lower cost of training as compared to Neural MT
- Applications: Machine Translation, Automated Bilingual Dictionary Generation, Cross-lingual Information Retrieval, *etc.*

Types of Cross-lingual Embeddings

- Based on the underlying approaches:
  - Monolingual mapping
  - Cross-lingual training
  - Joint optimization
- Based on the resource used:
  - Word-aligned data
  - Sentence-aligned data
  - Document-aligned data
  - Lexicon
  - No parallel data

Monolingual Mapping

- Learning in two step:
  - Train separate embeddings w_e and w_f on large monolingual corpora of corresponding languages e and f
  - Learn transformations $g1$ and $g2$ such that $w_e = g1(w_f)$ and $w_f = g2(w_e)$
- Transformations learnt using bilingual word mappings (lexicon)

Monolingual Mapping (contd.)

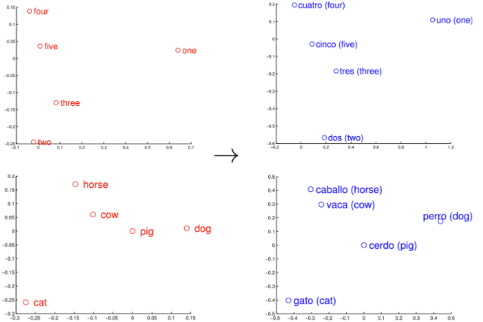
- Linear Projection proposed by Mikolov et al. (2013a)

- Learn matrix W s.t

$$w_e \approx W.w_f$$

which minimizes

$$\sum_{i=1}^{n} \|Ww_f - w_e\|^2$$



- We adapted this method for automatic synset linking in multilingual wordnets (accepted at GWC 2018)

Monolingual Mapping (Contd.)

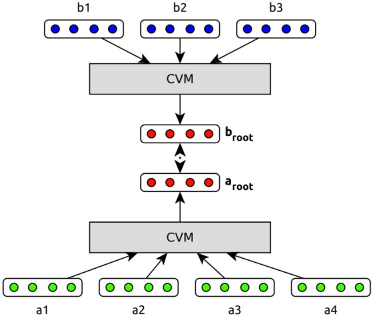
- Linear projection (Mikolov et al., 2013a): Lexicon
- Projection via CCA (Faruqui and Dyer, 2014b): Lexicon
- Alignment-based projection (Guo et al., 2015): Word-aligned data
- Adversarial auto-encoder (Barone, 2016): No parallel data

Cross Lingual Training

- Goal: optimizing cross-lingual objective
- Mainly rely on sentence alignments
- Require parallel corpus for training

Cross Lingual Training (contd.)

- Bilingual Compositional Sentence Model proposed by Hermann and Blunsom (2013)
- Train two models to produce sentence representations of aligned sentences in two languages
- Minimize distance between sentence representations of aligned sentences



Cross Lingual Training (contd.)

- Bilingual compositional sentence model (Hermann and Blunsom, 2013): Sentence-aligned data
- Distributed word alignment (Kočiskỳ et al., 2014): Sentence-aligned data
- Translation-invariant LSA (Huang et al., 2015): Lexicon
- Inverted Indexing on Wikipedia (Søgaard et al., 2015): Document-aligned data

Joint Optimization

- Jointly optimize both monolingual M and cross-lingual Ω constraints
- Objective: minimize $M_{l_1} + M_{l_2} + \lambda.\Omega_{l_1 \rightarrow l_2} + \Omega_{l_2 \rightarrow l_1}$
  where λ decides weightage of cross-lingual constraints

Joint Optimization (contd.)

- Multitask Language Model proposed by Klementiev et al. (2012):
  - Train neural language model (NNLM)
  - Jointly optimize monolingual maximum likelihood (M) with word alignment based MT regularization term (Ω)

Joint Optimization (contd.)

- Multi-task language model (Klementiev et al., 2012): Word-aligned data
- Bilingual skip-gram (Luong et al., 2015): Word-aligned data
- Bilingual bag-of-words without alignment (Gouws et al., 2015): Sentence-aligned data
- Bilingual sparse representations (Vyas and Carpuat, 2016): Word-aligned data

Word Embeddings
Interpretable Word Embeddings

Interpretability and Explainability

- A model is interpretable if a human can make sense out of it
- Example: Decision trees
- Interpretable models enable one to explain the performance of the system and tune it accordingly
- However, in practice, interpretable models generally perform poor compared to other systems

Interpretable Word Embeddings

- Dimensions interpretable by ordering words based on value

Word	d205
iguana	0.599371
bueller	0.584335
chimpanzee	0.577834
wasp	0.556845
chimp	0.553980
hamster	0.534810
giraffe	0.532316
unicorn	0.529533
caterpillar	0.528376
baboon	0.526324
gorilla	0.521590
tortoise	0.519941
sparrow	0.516842
lizard	0.515716
cockroach	0.505015
crocodile	0.491139
alligator	0.486275
moth	0.471682
kangaroo	0.469284
toad	0.463514

Word	d272
thigh	0.875286
knee	0.872282
shoulder	0.866209
elbow	0.857403
wrist	0.852959
ankle	0.851555
groin	0.841347
forearm	0.837988
leg	0.836661
pelvis	0.777564
neck	0.758420
spine	0.754774
torso	0.707458
hamstring	0.701921
buttocks	0.689092
knees	0.676485
ankles	0.658485
jaw	0.653126
biceps	0.650972
hips	0.647000

Examples from NNSE embeddings Murphy et al. (2012)

NNSE

- Non Negative Sparse Embeddings proposed by Murphy et al. (2012)
- Word embeddings interpretable and cognitively plausible
- Performs a mixture of topical and taxonomical semantics
- Computation
  - Dependency co-occurrence adjusted with PPMI (to normalize for word frequency) and reduced with sparse SVD
  - Document co-occurrence adjusted with PPMI and reduced with sparse SVD
  - Their union factorized using a variant of non-negative sparse coding
- Resulting word embeddings have both topical neighbors (*judge* is near to *prison*) and taxonomical neighbors (*judge* is near to *referee*)
- Code unavailable, embeddings available at `http://www.cs.cmu.edu/~bmurphy/NNSE/`

OIWE

- Online Interpretable Word Embeddings proposed by Luo et al. (2015)
- Main idea: apply sparsity to skip gram
- Achieve sparsity by setting to 0 any dimensions of a vector that falls below 0
- Propose two techniques to do this via gradient descent
- They outperform NNSE at word intrusion task
- Code available on Github at `https://github.com/SkTim/OIWE`

Evaluating Word Embeddings
Intrinsic Evaluation

Word Pair Similarity

- Evaluates generalizability of word embeddings
- One of the most widely used evaluations
- Many datasets available: WS353, RG65, MEN, SimLex, SCWS, *etc.*

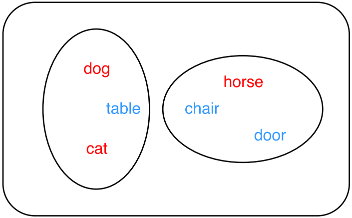
Word1	**Word2**	**Human Score**	**Model1 Score**	**Model2 Score**
street	street	10.00	1.0	1.0
street	avenue	8.88	0.04	0.38
street	block	6.88	0.14	0.26
street	place	6.44	0.21	0.18
street	children	4.94	-0.08	0.15
Spearman Correlation			**0.6**	**1.0**

Word Analogy task

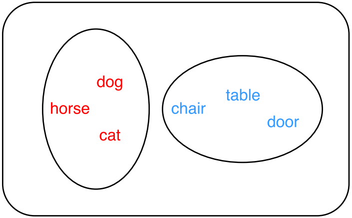
- Proposed by Mikolov et al. (2013b)
- Try to answer the question
  man is to woman as king is to ?
- Often discussed in media

Categorization

- Evaluates the ability of embeddings to form proper clusters
- Given sets of words with different labels, try to cluster them, and check the correspondence between clusters and sets.
- The *purer* the cluster, the better is the embeddings
- Datasets available: Bless, Battig, *etc.*



E1



E2

Word Intrusion Detection

- Proposed by Murphy et al. (2012)
- Provides a way to interpret dimensions
- Most approaches do not report results on this task
  - Experiments done by us suggest many of them are not interpretable

Word Intrusion Detection (contd.)

- The approach:
  1. Select a dimension
  2. Reverse sort all vectors based on this dimension
  3. Select top 5 words
  4. Select a word, which is in bottom half of this list, and is in top 10 percentile in some other columns
  5. Give a random permutation of these 6 words to a human evaluator
     - Example: {bathroom, closet, attic, balcony, quickly, toilet}
  6. Check precision

Evaluating Word Embeddings

Extrinsic Evaluation

Extrinsic Evaluations

- Evaluating word embeddings on downstream NLP tasks such as Part of speech tagging, Named Entity Recognition, *etc.*
- Makes more sense as we ultimately want to use embeddings for such tasks
- However, performance does not solely rely on embeddings
  - Improvement/Degradation could be due to other factors such as network architecture, hyperparameters, *etc.*
- If an embedding E_1 is better than another embedding E_2 when used with some network architecture for NER, does that mean E_1 will be better for all architectures of NER?

Evaluations on Unified Architectures

- Unified architectures such as Collobert and Weston (2008b) used for extrinsic evaluations
- For different tasks, the architecture remains same, except the last layer, where the output neurons are changed according to the task at hand
- If an embedding E_1 is better than another embedding E_2 on all tasks on such a unified architecture, then we can expect it to be truly better

Evaluating Word Embeddings
Evaluation Frameworks

WordVectors.org

- Proposed by Faruqui and Dyer (2014a)
- A web interface for evaluating a collection of word pair similarity datasets on your embeddings available at `http://wordvectors.org/`
- Also provides visualization for common sets of words like (Male,Female) and (Antonym,Synonym) pairs

VecEval

- Proposed by Nayak et al. (2016)
- A web based tool for performing extrinsic evaluations `http://www.veceval.com/`
- Claimed to support six different tasks: POS, NER, Chunking, Sentiment Analysis, Natural Language Inference, Question Answering
- Has never worked for me
- Web interface ~~no longer available~~ inactive, code available on Github at `https://github.com/NehaNayak/veceval`

Anago

- A Keras implementation of sequence labelling based on Lample et al. (2016)'s architecture
- Can perform POS, NER, SRL, *etc.*
- Used in our lab for extrinsic evaluation
- Code available on Github at `https://github.com/Hironsan/anago`

Evaluating Word Embeddings

Visualizing Word Embeddings

Visualizing Word Embeddings

- Various ways to visualize word embeddings: PCA, Isomap, tSNE, *etc.*, available in scikit-learn

```
from sklearn import decomposition, manifold
vis = decomposition.TruncatedSVD(n_components=2) - PCA
E_vis = vis.fit_transform( E)
plot E_vis here
```

- Check out `http://scikit-learn.org/stable/auto_examples/manifold/plot_lle_digits.html` for many methods applied to MNIST visualization

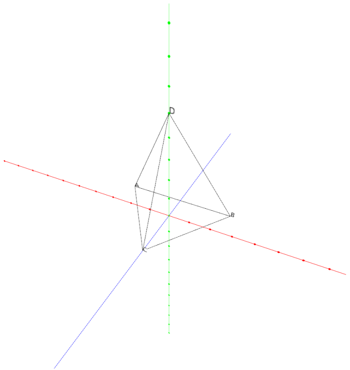
Related Work

- Baroni et al. (2014): Neural word embeddings are better than traditional methods such as LSA, HAL, RI (Landauer and Dumais, 1997; Lund and Burgess, 1996b; Sahlgren, 2005)
- Levy et al. (2015): Superiority of neural word embeddings not due to the embedding algorithm, but due to **certain design choices and hyperparameters optimizations**
  - Varies other hyperparameters; keeps number of dimensions = 500
- Schnabel et al. (2015); Zhai et al. (2016); Ghannay et al. (2016): No justification for chosen number of dimensions in their evaluations
- Melamud et al. (2016): Optimal number of dimensions different for different evaluations of word embeddings

Why Dimensions matter?: A Practical Example

- Various app developers want to utilize word embeddings
- Example memory limit for app: 200 MB
- Size of Google Pre-trained vectors file: 3.4 GB
- Natural thought process: decrease dimensions
  - To what value? 100? 50? 20?
- Depends on the words/entities we want to place in the space

Number of Dimensions and Equidistant points



Number of Dimensions and Equidistant points

- Number of dimensions of a vector space imposes a restriction on the number of equidistant points it can have
- Given that distance is euclidean, if the number of dimensions $\lambda = N$, then maximum number of equidistant points E in the corresponding space is $N + 1$ (Swanepoel, 2004)
- Given that distance is cosine, no closed form solution exists

Dimensions λ and max. no. of equiangular lines E(Barg and Yu, 2014)

λ	E	λ	E
3	6	18	61
4	6	19	76
5	10	20	96
6	16	21	126
7<=n<=13	28	22	176
14	30	23	276
15	36	24<=n<=41	276
16	42	42	288
17	51	43	344

Objective

Problem Statement

Does the number of pairwise equidistant words enforce a lower bound on the number of dimensions for word embeddings?

- 'Equidistance' determined using co-occurrence matrix
- Plan of Action:
  - Verify using a toy corpus
  - Evaluate on actual corpus

Motivation (1/4)

- Consider the following toy corpus:

  <>I like cats <>I love dogs <>I hate rats <>I rate bats <>
- Corresponding co-occurrence matrix:

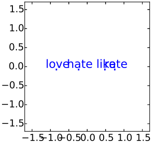
word	<>	I	like	love	hate	rate	rats	cats	dogs	bats
like	0	1	0	0	0	0	0	1	0	0
love	0	1	0	0	0	0	0	0	1	0
hate	0	1	0	0	0	0	1	0	0	0
rate	0	1	0	0	0	0	0	0	0	1

- Distance between any pair of words $= \sqrt{2}$
- The words form a regular tetrahedron

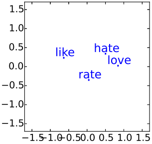
Motivation (2/4)

Mean and Std Dev of Mean of a point's distance with other points

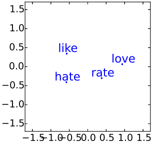
Dimension	Mean	Stddev
1	0.94	0.94
2	1.77	0.80
3	2.63	0.10



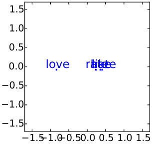
1d(Before)



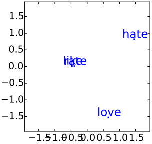
2d(Before)



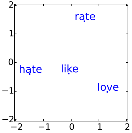
3d(Before)



1d(After)



2d(After)



3d(After)

Motivation (3/4)

Hypothesis:

- If the learning algorithm of word embeddings does not get enough dimensions, then it will fail to uphold the equality constraint
  - Standard deviation of the mean of all pairwise distances will be higher
- As we increase the dimension, the algorithm will get more degrees of freedom to model the equality constraint in a better way
  - There will be statistically significant changes in the standard deviation
- Once the lower bound of dimensions is reached, the algorithm gets enough degrees of freedom.
  - From this point onwards, even if we increase dimensions, there will not be any statistically significant difference in the standard deviation

Motivation (4/4)

Dim	$\overline{\sigma}$	P-value	Dim	$\overline{\sigma}$	P-value
7	0.358		12	0.154	0.0058
8	0.293	0.0020	13	0.111	0.0001
9	0.273	0.0248	14	0.044	0.0001
10	0.238	0.0313	15	0.047	**0.3096**
11	0.189	0.0013	16	0.054	**0.1659**

Avg standard deviation ($\overline{\sigma}$) for 15 pairwise equidistant words (along with two tail p-values of Welch's unpaired t-test for statistical significance)

Approach (1/5)

1. Compute the word $\times$ word co-occurrence matrix from the corpus

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

word	<>	I	like	love	hate	rate	rats	cats	dogs	bats
<>	0	4	0	0	0	0	1	1	1	1
I	4	0	1	1	1	1	0	0	0	0
like	0	1	0	0	0	0	0	1	0	0
love	0	1	0	0	0	0	0	0	1	0
hate	0	1	0	0	0	0	1	0	0	0
rate	0	1	0	0	0	0	0	0	0	1
rats	1	0	0	0	1	0	0	0	0	0
cats	1	0	1	0	0	0	0	0	0	0
dogs	1	0	0	1	0	0	0	0	0	0
bats	1	0	0	0	0	1	0	0	0	0

Approach (2/5)

2. Create the corresponding word $\times$ word cosine similarity matrix

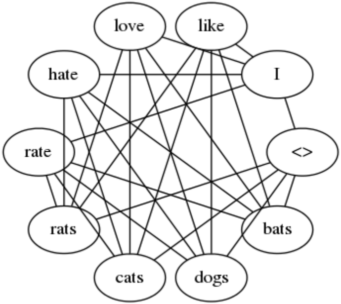
<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

	<>	I	like	love	hate	rate	rats	cats	dogs	bats
<>	1.0	0.0	0.8	0.8	0.8	0.8	0.0	0.0	0.0	0.0
I	0.0	1.0	0.0	0.0	0.0	0.0	0.8	0.8	0.8	0.8
like	0.8	0.0	1.0	0.5	0.5	0.5	0.0	0.0	0.0	0.0
love	0.8	0.0	0.5	1.0	0.5	0.5	0.0	0.0	0.0	0.0
hate	0.8	0.0	0.5	0.5	1.0	0.5	0.0	0.0	0.0	0.0
rate	0.8	0.0	0.5	0.5	0.5	1.0	0.0	0.0	0.0	0.0
rats	0.0	0.8	0.0	0.0	0.0	0.0	1.0	0.5	0.5	0.5
cats	0.0	0.8	0.0	0.0	0.0	0.0	0.5	1.0	0.5	0.5
dogs	0.0	0.8	0.0	0.0	0.0	0.0	0.5	0.5	1.0	0.5
bats	0.0	0.8	0.0	0.0	0.0	0.0	0.5	0.5	0.5	1.0

Approach (3/5)

3. For each similarity value s_k, create a graph, where the words are nodes, and an edge between node i and node j if $\text{sim}(i, j) = s_k$

$<>$ I like cats $<>$ I love dogs $<>$ I hate rats $<>$ I rate bats $<>$

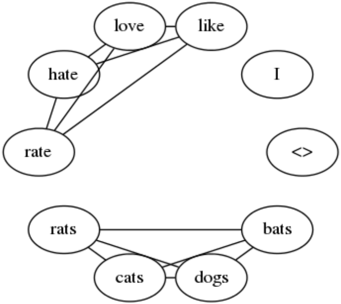


Sim=0.0

Approach (3/5)

3. For each similarity value s_k, create a graph, where the words are nodes, and an edge between node i and node j if sim(i, j) $= s_k$

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

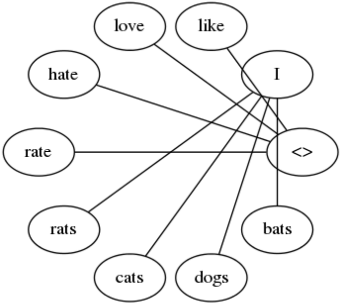


Sim=0.5

Approach (3/5)

3. For each similarity value s_k, create a graph, where the words are nodes, and an edge between node i and node j if $\mathrm{sim}(i, j) = s_k$

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

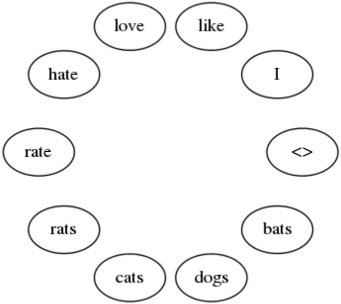


Sim=0.8

Approach (3/5)

3. For each similarity value s_k, create a graph, where the words are nodes, and an edge between node i and node j if $\text{sim}(i, j) = s_k$

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>



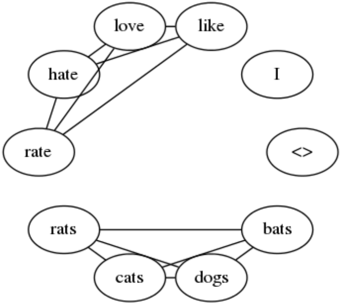
Sim=1.0

Approach (4/5)

4. Find maximum clique on this graph. The number of nodes in this clique is the maximum number of pairwise equidistant points E_k

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

Sim	E_k
0.5	4

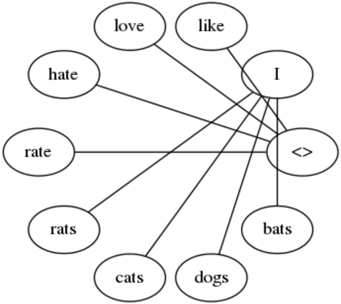


Approach (4/5)

4. Find maximum clique on this graph. The number of nodes in this clique is the maximum number of pairwise equidistant points E_k

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

Sim	E_k
0.5	4
0.8	0

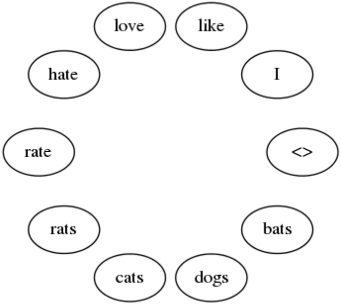


Approach (4/5)

4. Find maximum clique on this graph. The number of nodes in this clique is the maximum number of pairwise equidistant points E_k

<> I like cats <> I love dogs <> I hate rats <> I rate bats <>

Sim	E_k
0.5	4
0.8	0
1.0	0



Approach (5/5)

5. Reverse lookup E_k to get the number of dimension λ

$<>$ I like cats $<>$ I love dogs $<>$ I hate rats $<>$ I rate bats $<>$

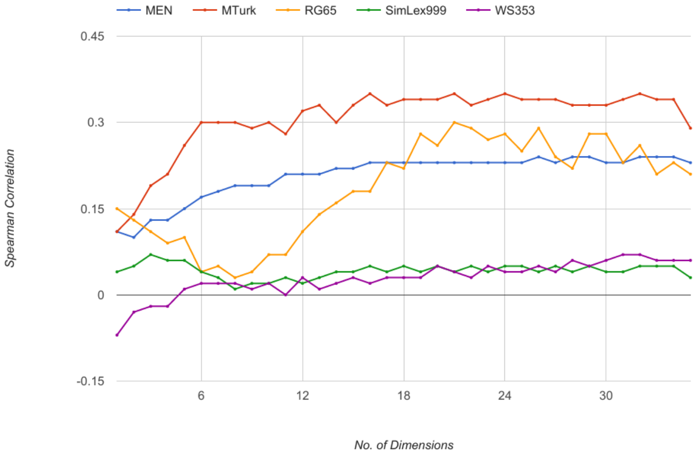
Sim	E_k
0.5	4
0.8	0
1.0	0
Max	4

λ	E	λ	E
3	6	18	61
4	6	19	76
5	10	20	96
6	16	21	126
7<=n<=13	28	22	176
14	30	23	276
15	36	24<=n<=41	276
16	42	42	288
17	51	43	344

Evaluation

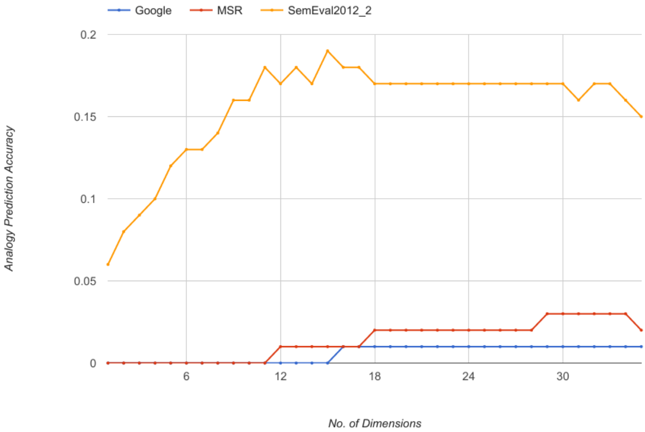
- Used Brown Corpus
- Found **19** as lower bound using our approach
- Context window: 1 to the left and 1 to the right
- Number of dimensions: 1 to 35
- 5 randomly initialized models for each configuration (average results reported)
- Intrinsic Evaluation
  - Word Pair Similarity: Predicting $sim(w_a, w_b)$ using corresponding word embeddings
  - Word Analogy: Finding missing w_d in the relation: a is to b as c is to d
  - Categorization: Checking the purity of clusters formed by word embeddings

Results



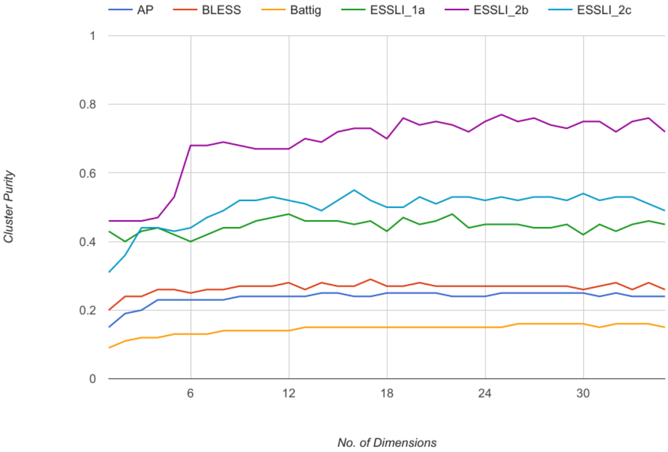
Performance for Word Pair Similarity task with respect to number of dimensions

Results



Performance for Word Analogy task with respect to number of dimensions

Results



Performance for Categorization task with respect to number of dimensions

Analysis

- Found lower bound consistent with experimental evaluation

Poltawa	snakestrike	burnings	Tsar's
miswritten	brows	maintained	South-East
far-famed	27%	non-dramas	octagonal
boatyards	U-2	Devol	mourners
Hearing	sideshow	third-story	upcoming
pram	dolphins	Croydon	neuromuscular
Gladius	pvt	littered	annoying
vuhranduh	athletes	eraser	provincialism
Daly	wreaths	villain	suspicious
nooks	fielder	belly	Gogol's
interchange	two-to-three	resemble	discounted
kidneys	Hangman's	commend	accordion
summarizing	optimality	Orlando	Leamington
swift	Taras-Tchaikovsky	puts	groomed
spit	firmer	rosy-fingered	Bechhofer
campfire	Tomas		

Set of pairwise equiangular points (vectors) from Brown corpus

Limitations

- The Max Clique finding component of the approach
  - Renders approach intractable for larger corpora
  - Need to find an alternative

Applications of Word Embeddings

Are Word Embeddings Useful for Sarcasm Detection?

Problem Statement

- Detect whether a sentence is sarcastic or not?
  - Especially among those sentences which do not contain sentiment bearing words
- Example: A woman needs a man just like a fish needs a bicycle

Motivation

- Similarity measure among word embeddings a proxy for measuring contextual incongruity
- Example: A woman needs a man just like a fish needs a bicycle

$$\textit{similarity(man,woman)} = \textit{0.766}$$
$$\textit{similarity(fish,bicycle)} = \textit{0.131}$$

- Imbalance in similarities above an indication of contextual incongruity

Approach

- Gist of the approach is adding similarity of word embeddings based features, such as
  - Maximum similarity between all pairs of words in a sentence
  - Minimum similarity between all pairs of words in a sentence

Evaluation

1. **Liebrecht et al. (2013)**: They consider unigrams, bigrams and trigrams as features.
2. **González-Ibánez et al. (2011)**: Two sets of features: unigrams and dictionary-based.
3. **Buschmeier et al. (2014)**:
   - Hyperbole (captured by 3 positive or negative words in a row)
   - Quotation marks and ellipsis
   - Positive/Negative Sentiment words followed by an exclamation or question mark
   - Positive/Negative Sentiment Scores followed by ellipsis ('...')
   - Punctuation, Interjections, and Laughter expressions.
4. **Joshi et al. (2015)**: In addition to unigrams, they use features based on implicit and explicit incongruity
   - Implicit incongruity features - patterns with implicit sentiment , extracted in a pre-processing step.
   - Explicit incongruity features - number of sentiment flips, length of positive and negative sub-sequences and lexical polarity.

Results

Word Embedding	Average F-score Gain
LSA	0.453
Glove	0.651
Dependency	1.048
Word2Vec	1.143

Average gain in F-scores for the four types of word embeddings; These values are computed for a subset of these embeddings consisting of words common to all four

Applications of Word Embeddings
Iterative Unsupervised Most Frequent Sense Detection using Word Embeddings

WordNet

- Groups synonymous words into *synsets*
- Synset example:
  - Synset ID: 02139199
  - Synset Members: { bat, chiropteran }
  - Gloss: nocturnal mouselike mammal with forelimbs modified to form membranous wings and anatomical adaptations for echolocation by which they navigate
  - Example: Bats are creatures of the night.
- Relations with other synsets (hypernym/hyponym: parent/child, meronym/holonym: part/whole)

Introduction

- Word Sense Disambiguation (WSD) : one of the relatively hard problems in NLP
  - Both supervised and unsupervised ML explored in literature
- Most Frequent Sense (MFS) baseline: strong baseline for WSD
  - Given a WSD problem instance, simply assign the most frequent sense of that word
- Ignores context
- Really strong results
  - Due to skew in sense distribution of data
- Computing MFS:
  - Trivial for sense-annotated corpora, which is not available in large amounts.
  - Need to learn from raw data

Problem Statement

Problem Statement

Given a raw corpus, estimate most frequent sense of different words in that corpus

- Bhingardive et al. (2015) showed that pretrained word embeddings can be used to compute most frequent sense
- Our work further strengthens the claim by Bhingardive et al. (2015) that word embeddings indeed capture most frequent sense
- Our approach outperforms others at the task of MFS extraction
- To compute MFS using our approach:
  1. Train word embeddings on the raw corpus.
  2. Apply our approach on the trained word embeddings.

Intuition

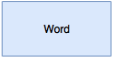
- Strive for consistency in assignment of senses to maintain semantic congruity
- Example:
  - If *cricket* and *bat* co-occur a lot, then *cricket* taking *insect* sense and *bat* taking reptile sense is less likely

Intuition

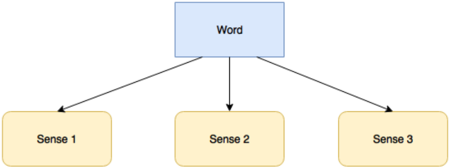
- Strive for consistency in assignment of senses to maintain semantic congruity
- Example:
  - If *cricket* and *bat* co-occur a lot, then *cricket* taking *insect* sense and *bat* taking reptile sense is less likely
  - If *cricket* and *bat* co-occur a lot, and *cricket*'s MFS is *sports*, then *bat* taking reptile sense is extremely unlikely
- Key point: solve easy words, then use them for difficult words

  In other words, iterate over degree of polysemy from 2 onward

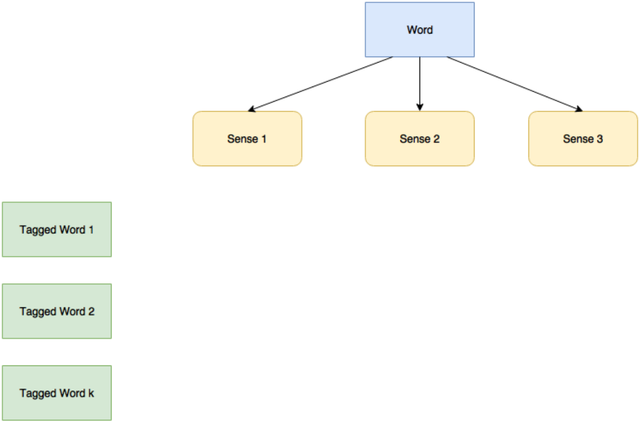
Algorithm



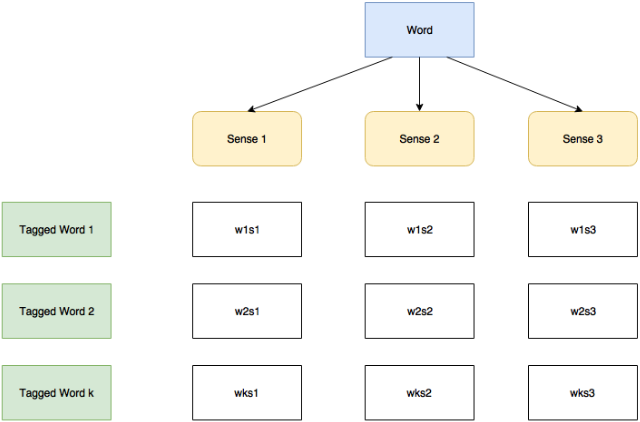
Algorithm



Algorithm


Word
Sense 1
Sense 2
Sense 3
Tagged Word 1
Tagged Word 2
Tagged Word k

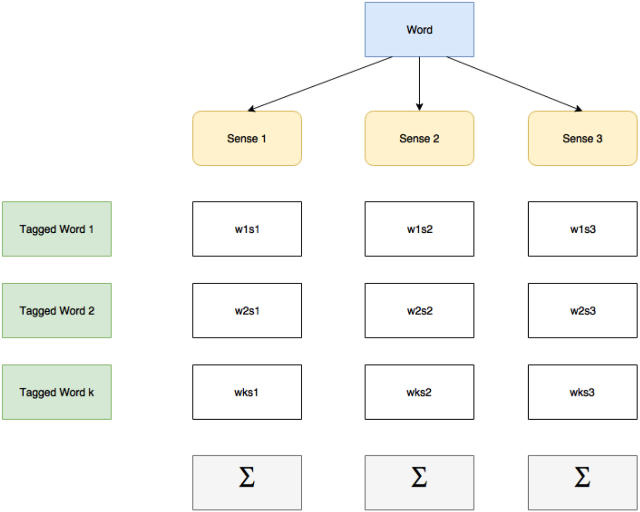
Algorithm


Word
Sense 1
Sense 2
Sense 3
Tagged Word 1
Tagged Word 2
Tagged Word k
w1s1
w1s2
w1s3
w2s1
w2s2
w2s3
wks1
wks2
wks3

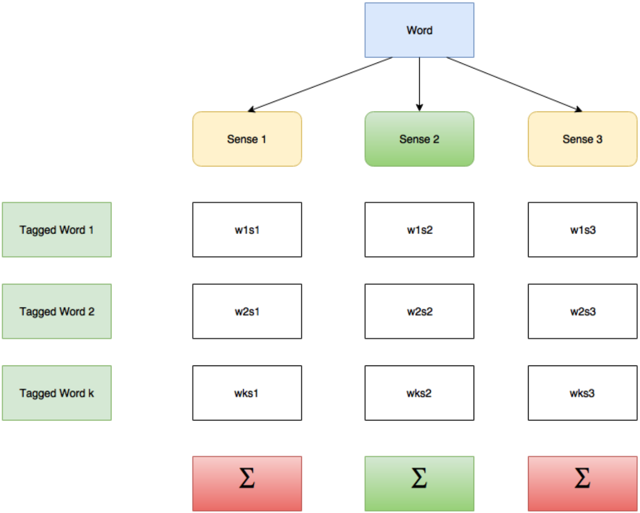
Algorithm

- $w_i s_j$ is vote for s_j due to w_i
- Two components
  - Wordnet similarity between mfs(w_i) and s_j
  - Embedding space similarity between w_i and current word

Algorithm


Word
Sense 1
Sense 2
Sense 3
Tagged Word 1
Tagged Word 2
Tagged Word k
w1s1
w1s2
w1s3
w2s1
w2s2
w2s3
wks1
wks2
wks3
Σ
Σ
Σ

Algorithm


Word
Sense 1
Sense 2
Sense 3
Tagged Word 1
Tagged Word 2
Tagged Word k
w1s1
w1s2
w1s3
w2s1
w2s2
w2s3
wks1
wks2
wks3
Σ
Σ
Σ

Parameters

- K
- Similarity Measure
- Unweighted (no vector space component) vs. Weighted

Evaluation

- Two setups:
  - Evaluating MFS as solution for WSD
  - Evaluating MFS as a classification task

MFS as solution for WSD

Method	Senseval2	Senseval3
Bhingardive(reported)	52.34	43.28
SemCor(reported)	59.88	65.72
Bhingardive	48.27	36.67
Iterative	63.2	56.72
SemCor	67.61	71.06

Accuracy of WSD using MFS (Nouns)

MFS as solution for WSD (contd.)

Method	Senseval2	Senseval3
Bhingardive(reported)	37.79	26.79
Bhingardive(optimal)	43.51	33.78
Iterative	48.1	40.4
SemCor	60.03	60.98

Accuracy of WSD using MFS (All Parts of Speech)

MFS as classification task

Method	Nouns	Adjectives	Adverbs	Verbs	Total
Bhingardive	43.93	81.79	46.55	37.84	58.75
Iterative	48.27	80.77	46.55	44.32	61.07

Percentage match between predicted MFS and WFS

MFS as classification task (contd.)

	Nouns (49.20)	**Verbs (26.44)**	**Adjectives (19.22)**	**Adverbs (5.14)**	**Total**
Bhingardive	29.18	25.57	26.00	33.50	27.83
Iterative	35.46	31.90	30.43	47.78	34.19

Percentage match between predicted MFS and true SemCor MFS. Note that numbers in column headers indicate what percent of total words belong to that part of speech

Analysis

- Better than Bhingardive et al. (2015); not able to beat SemCor and WFS.
  - There are words for which WFS doesn't give *proper* dominant sense. Consider the following examples:
    - *tiger* - an audacious person
    - *life* - characteristic state or mode of living (social life, city life, real life)
    - *option* - right to buy or sell property at an agreed price
    - *flavor* - general atmosphere of place or situation
    - *season* - period of year marked by special events
  - Tagged words ranking very low to make a significant impact. For example:
    - While detecting MFS for a bisemous word, the first monosemous neighbour actually ranks 1101
    - *i.e.* a 1000 polysemous words are closer than this monosemous word.
    - Monosemous word may not be the one who can influence the MFS.

Summary

- Proposed an iterative approach for unsupervised most frequent sense detection using word embeddings
- Similar trends, yet better overall results from Bhingardive et al. (2015)
- Future Work
  - Apply approach to other languages

Conclusion

- Discussed why we need word embeddings
- Briefly looked at classical word embeddings
- Discussed a few cross-lingual word embeddings and interpretable word embeddings
- Mentioned evaluation mechanisms and tools
- Argued on existence of lower bounds for number of dimensions of word embeddings
- Discussed some in-house applications

Thank You

References

Barg, A. and Yu, W.-H. (2014). New bounds for equiangular lines. *Contemporary Mathematics*, 625:111–121.

Barone, A. V. M. (2016). Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. *arXiv preprint arXiv:1608.02996*.

Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247. Association for Computational Linguistics.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

References

Bhingardive, S., Singh, D., V, R., Redkar, H., and Bhattacharyya, P. (2015). Unsupervised most frequent sense detection using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1238–1243, Denver, Colorado. Association for Computational Linguistics.

Buschmeier, K., Cimiano, P., and Klinger, R. (2014). An impact analysis of features in a classification approach to irony detection in product reviews. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 42–49.

References

Collobert, R. and Weston, J. (2008a). A unified architecture for natural language processing: deep neural networks with multitask learning. In Cohen, W. W., McCallum, A., and Roweis, S. T., editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Collobert, R. and Weston, J. (2008b). A unified architecture for natural language processing: deep neural networks with multitask learning. In Cohen, W. W., McCallum, A., and Roweis, S. T., editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285. ACM.

References

Faruqui, M. and Dyer, C. (2014a). Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, USA. Association for Computational Linguistics.

Faruqui, M. and Dyer, C. (2014b). Improving vector space word representations using multilingual correlation. Association for Computational Linguistics.

Ghannay, S., Favre, B., Estéve, Y., and Camelin, N. (2016). Word embedding evaluation and combination. In Chair), N. C. C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

References

González-Ibánez, R., Muresan, S., and Wacholder, N. (2011). Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 581–586. Association for Computational Linguistics.

Gouws, S., Bengio, Y., and Corrado, G. (2015). Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 748–756.

Guo, J., Che, W., Yarowsky, D., Wang, H., and Liu, T. (2015). Cross-lingual dependency parsing based on distributed representations. In *ACL (1)*, pages 1234–1244.

Harris, Z. S. (1970). *Distributional structure*. Springer.

References

Hermann, K. M. and Blunsom, P. (2013). Multilingual distributed representations without word alignment. *arXiv preprint arXiv:1312.6173*.

Huang, K., Gardner, M., Papalexakis, E., Faloutsos, C., Sidiropoulos, N., Mitchell, T., Talukdar, P. P., and Fu, X. (2015). Translation invariant word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1084–1088.

Joshi, A., Sharma, V., and Bhattacharyya, P. (2015). Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 757–762.

References

Klementiev, A., Titov, I., and Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. *Proceedings of COLING 2012*, pages 1459–1474.

Kočiskỳ, T., Hermann, K. M., and Blunsom, P. (2014). Learning bilingual word representations by marginalizing alignments. *arXiv preprint arXiv:1405.0947*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *PSYCHOLOGICAL REVIEW*, 104(2):211–240.

References

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 302–308.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Liebrecht, C., Kunneman, F., and van den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets# not. *Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA*.

References

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304.

Lund, K. and Burgess, C. (1996a). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Lund, K. and Burgess, C. (1996b). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

References

Luo, H., Liu, Z., Luan, H., and Sun, M. (2015). Online learning of interpretable word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1687–1692.

Luong, T., Pham, H., and Manning, C. D. (2015). Bilingual word representations with monolingual quality in mind. In *VS@ HLT-NAACL*, pages 151–159.

Melamud, O., McClosky, D., Patwardhan, S., and Bansal, M. (2016). The role of context types and dimensionality in learning word embeddings. In Knight, K., Nenkova, A., and Rambow, O., editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1030–1040. The Association for Computational Linguistics.

References

Mikolov, T., Le, Q. V., and Sutskever, I. (2013a). Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Murphy, B., Talukdar, P., and Mitchell, T. (2012). *Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding*, pages 1933–1949. Association for Computational Linguistics.

References

Nayak, N., Angeli, G., and Manning, C. D. (2016). Evaluating word embeddings using a representative suite of practical tasks. *ACL 2016*, page 19.

Patel, K., Patel, D., Golakiya, M., Bhattacharyya, P., and Birari, N. (2017). Adapting pre-trained word embeddings for use in medical coding. In *BioNLP 2017*, pages 302–306, Vancouver, Canada,. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633.

References

Sahlgren, M. (2005). An introduction to random indexing. In *In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Institutionen för lingvistik.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 298–307.

Schütze, H. (1992). Dimensions of meaning. In *Supercomputing'92., Proceedings*, pages 787–796. IEEE.

References

Søgaard, A., Agić, Ž., Alonso, H. M., Plank, B., Bohnet, B., and Johannsen, A. (2015). Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*.

Swanepoel, K. J. (2004). Equilateral sets in finite-dimensional normed spaces. In *Seminar of Mathematical Analysis*, volume 71, pages 195–237. Secretariado de Publicationes, Universidad de Sevilla, Seville.

Vyas, Y. and Carpuat, M. (2016). Sparse bilingual word representations for cross-lingual lexical entailment. In *HLT-NAACL*, pages 1187–1197.

References

Zhai, M., Tan, J., and Choi, J. D. (2016). Intrinsic and extrinsic evaluations of word embeddings. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 4282–4283. AAAI Press.

Web References

- `http://www.indiana.edu/~gasser/Q530/Notes/representation.html`