StarSpace: Embed All The Things!

Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes and Jason Weston
Facebook AI
(2017)

Presenter: Derrick Blakely

Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

Outline

1. Background: Language Models and Previous Embedding Algos
2. Motivation of StarSpace
3. What is StarSpace? What is new about it?
4. Results
5. Conclusions

Outline

1. Background: Language Models and Previous Embedding Algos
2. Motivation of StarSpace
3. What is StarSpace? What is new about it?
4. Results
5. Conclusions

Background: Neural Languages Models

$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, ..., w_{t-n+1}]$$

Background: Neural Language Models

$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}]$$

- Each w_t parameterized with a set of values in a vector

Background: Neural Language Models

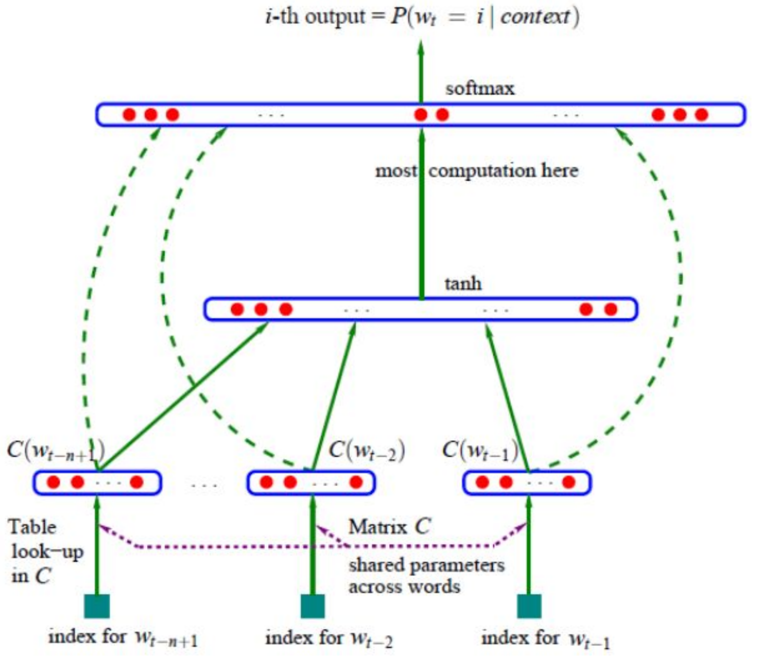
$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, ..., w_{t-n+1}]$$

- Each w_t parameterized with a set of values in a vector
- Bengio, 2003 - neural language model:
- Learning word representations stored lookup table/matrix

Background: Neural Language Models

$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}]$$

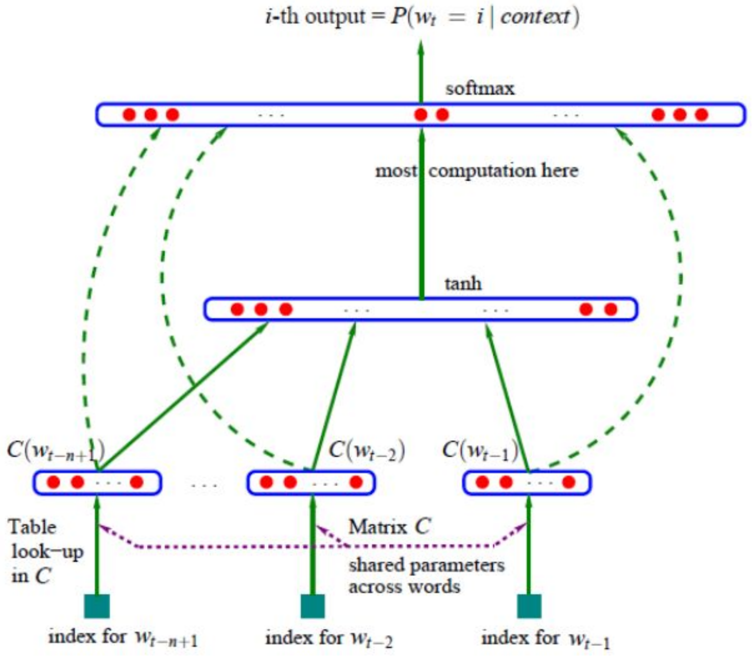
- Each w_t parameterized with a set of values in a vector
- Bengio, 2003 - neural language model:
- Learning word representations stored lookup table/matrix



Background: Neural Language Models

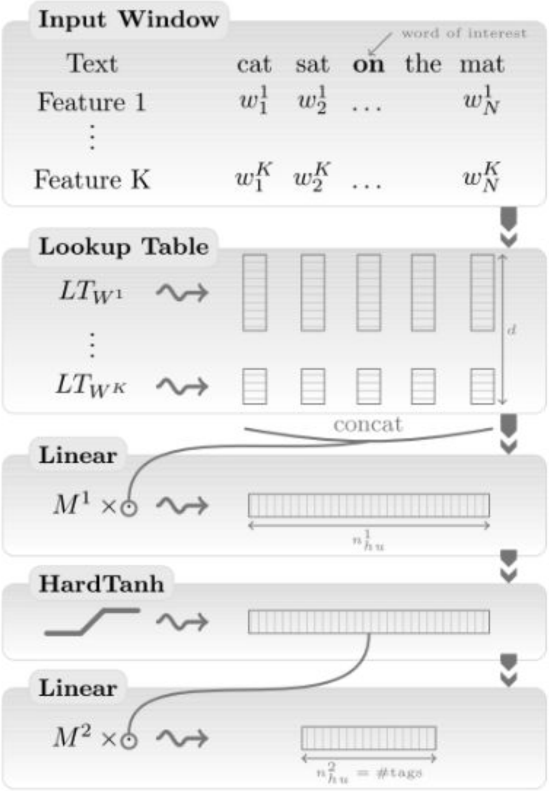
$$Pr[w|context] = Pr[w_t|w_{t-1}, w_{t-2}, \ldots, w_{t-n+1}]$$

- Impractical--extremely expensive fully-connected softmax layer
- Learned embeddings not transferable to other tasks



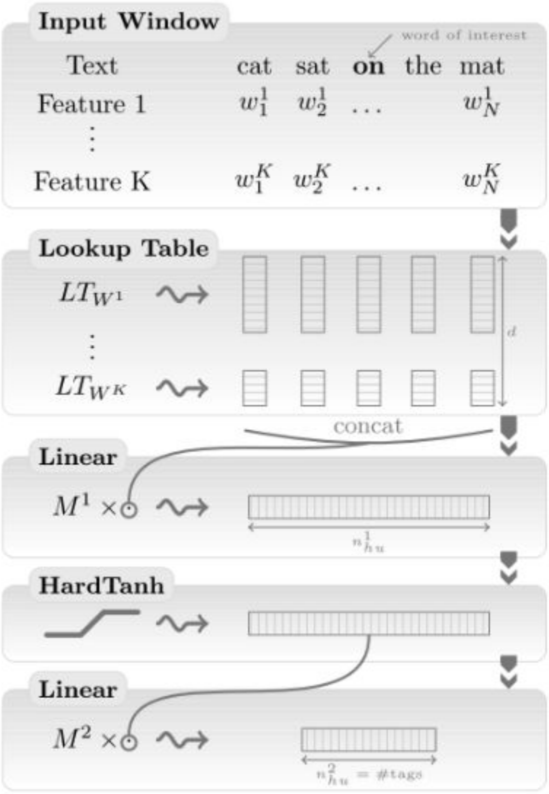
Background: Collobert & Weston (2008, 2011)

- Improved the objective function and removed expensive softmax layer



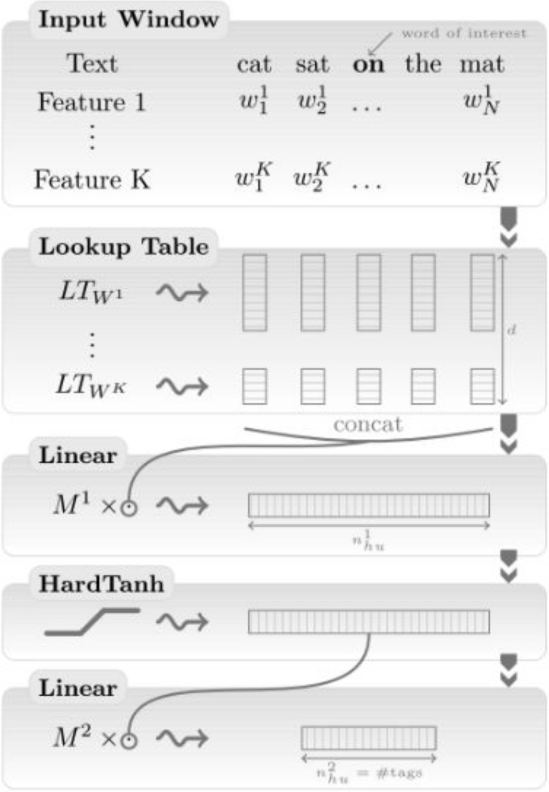
Background: Collobert & Weston (2008/2011)

- Improved the objective function and removed expensive softmax layer
- CNNs + tagging ➜ semantic embeddings



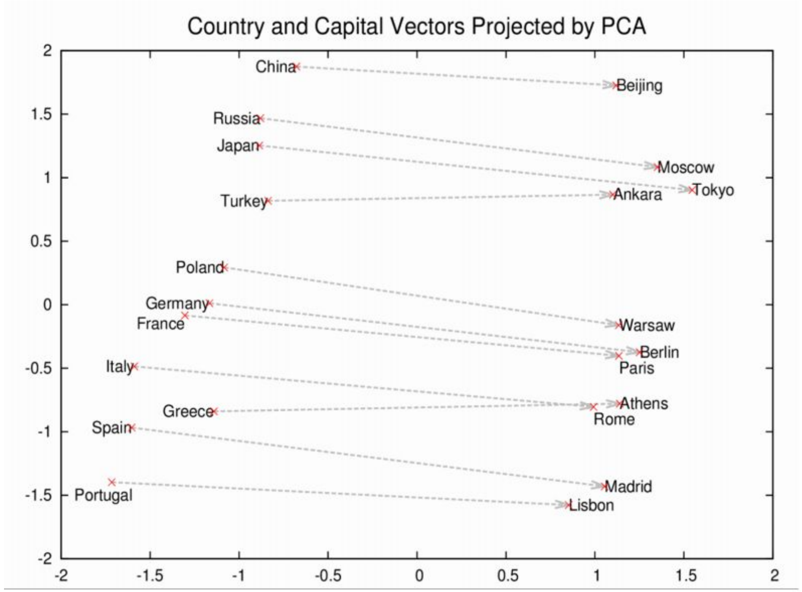
Background: Collobert & Weston (2008/2011)

- Improved the objective function and removed expensive softmax layer
- CNNs + tagging ➔ semantic embeddings
- Showed that learned embeddings could be useful for downstream tasks



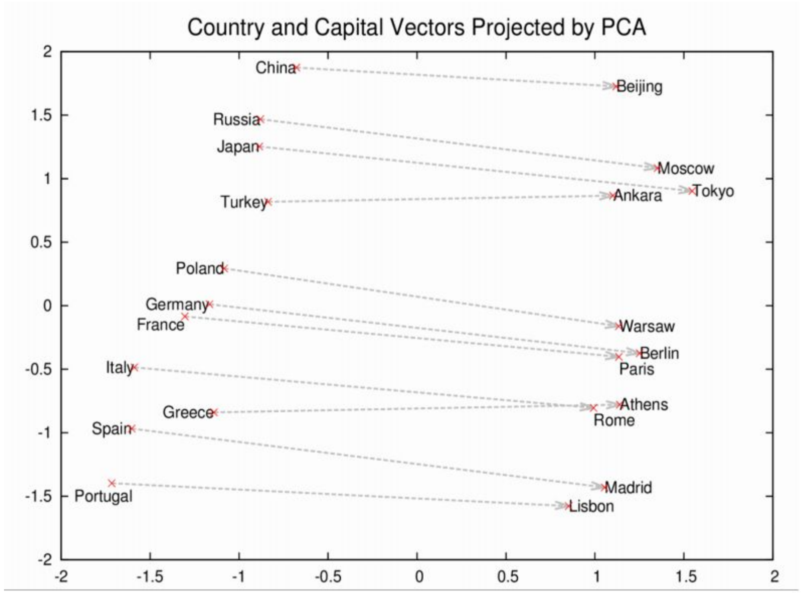
Popular Embedding Tools

- Word2vec (Mikolov, 2013)
- GloVe (Pennington, 2014)
- fastTest (Facebook, 2015)



Popular Embedding Tools

- Word2vec (Mikolov, 2013)
- GloVe (Pennington, 2014)
- fastTest (Facebook, 2015)



Common Issues:

- Slow fully-connected layers
- Limited to text sequences
- Can we embed, say, documents and labels in a common vector space?

Outline

1. Background: Language Models and Previous Embedding Algos
2. Motivation of StarSpace
3. What is StarSpace? What is new about it?
4. Results
5. Conclusions

Motivation of StarSpace

- Improve upon word2vec, fastText, and GloVe

Motivation of StarSpace

- Improve upon word2vec, fastText, and GloVe
- Generalizable ML: “Embed all the things”--not just text
  - Documents, words, sentences, labels, users, items to recommend to users, images

Motivation of StarSpace

- Improve upon word2vec, fastText, and GloVe
- Generalizable ML: “Embed all the things”--not just text
    - Documents, words, sentences, labels, users, items to recommend to users, images
- Embed entities of “Type A” with related entities of “Type B”

Motivation of StarSpace

- Improve upon word2vec, fastText, and GloVe
- Generalizable ML: “Embed all the things”--not just text
  - Documents, words, sentences, labels, users, items to recommend to users, images
- Embed entities of “Type A” with related entities of “Type B”
- Provide good (not necessarily *best*) performance for many tasks

Motivation of StarSpace

- Improve upon word2vec, fastText, and GloVe
- Generalizable ML: “Embed all the things”--not just text
  - Documents, words, sentences, labels, users, items to recommend to users, images
- Embed entities of “Type A” with related entities of “Type B”
- Provide good (not necessarily *best*) performance for many tasks
- StarSpace can be a goto baseline; tool you can try out on lots of problems

Outline

1. Background: Language Models and Previous Embedding Algos
2. Motivation of StarSpace
3. What is StarSpace? What is new about it?
4. Results
5. Conclusions

Entities in StarSpace

- Entity: words, sentences, documents, users, images, labels, etc.

Entities in StarSpace

- Entity: words, sentences, documents, users, images, labels, etc.
- Old way: words represented as a single word ID
- Raw sentence: [huge iceberg in Greenland] ➔ [60, 100, 4, 55]

Entities in StarSpace

- Entity: words, sentences, documents, users, images, labels, etc.
- Old way: words represented as a single word ID
- Raw sentence: [huge iceberg in Greenland] → [60, 100, 4, 55]
- $\mathrm{Embedding}(w_i) = \mathrm{LookupTable}[i] = [\Theta_{i1}, \Theta_{i2}, \ldots, \Theta_{i300}]^T$

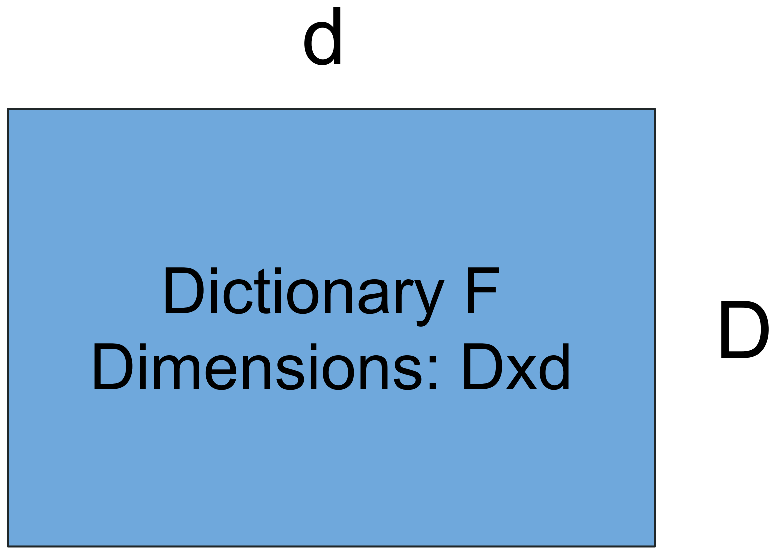
Entities in StarSpace

- Entity: words, sentences, documents, users, images, labels, etc.
- Old way: words represented as a single word ID
- Raw sentence: [huge iceberg in Greenland] → [60, 100, 4, 55]
- $\text{Embedding}(w_i) = \text{LookupTable}[i] = [\Theta_{i1}, \Theta_{i2}, \ldots, \Theta_{i300}]^T$
- StarSpace: *entities* are bags-of-features (sets of feature ID’s)

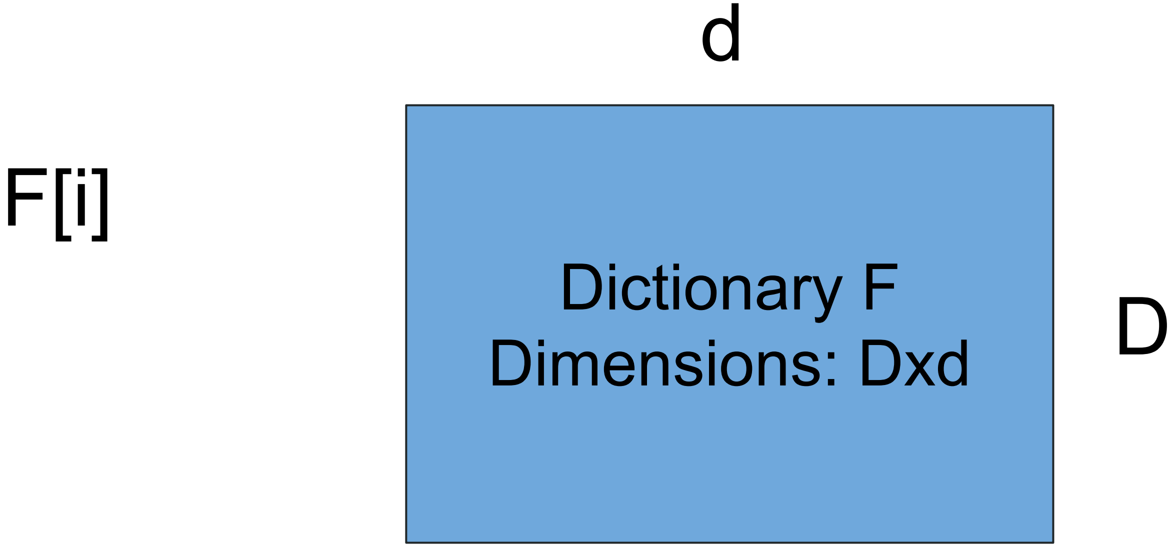
Entities in StarSpace

- Entity: words, sentences, documents, users, images, labels, etc.
- Old way: words represented as a single word ID
- Raw sentence: [huge iceberg in Greenland] ➔ [60, 100, 4, 55]
- $\text{Embedding}(w_i) = \text{LookupTable}[i] = [\Theta_{i1}, \Theta_{i2}, \ldots, \Theta_{i300}]^T$
- StarSpace: *entities* are bags-of-features (sets of feature ID's)
- Entity a = [60, 100, 4, 55]
- Embedding(a) = LookupTable[60] + … + LookupTable[55]

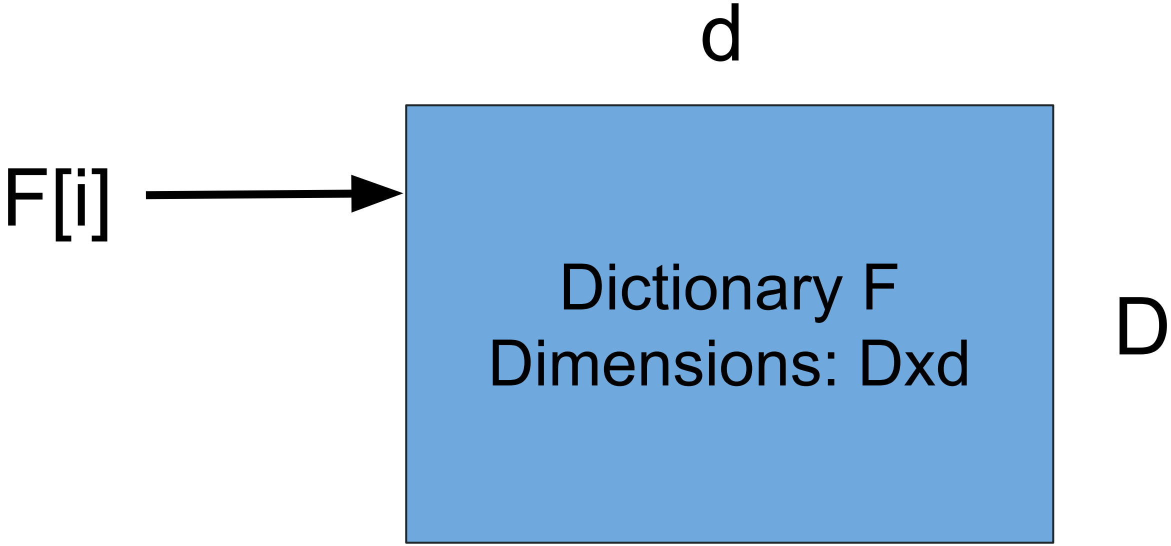
Entities in StarSpace



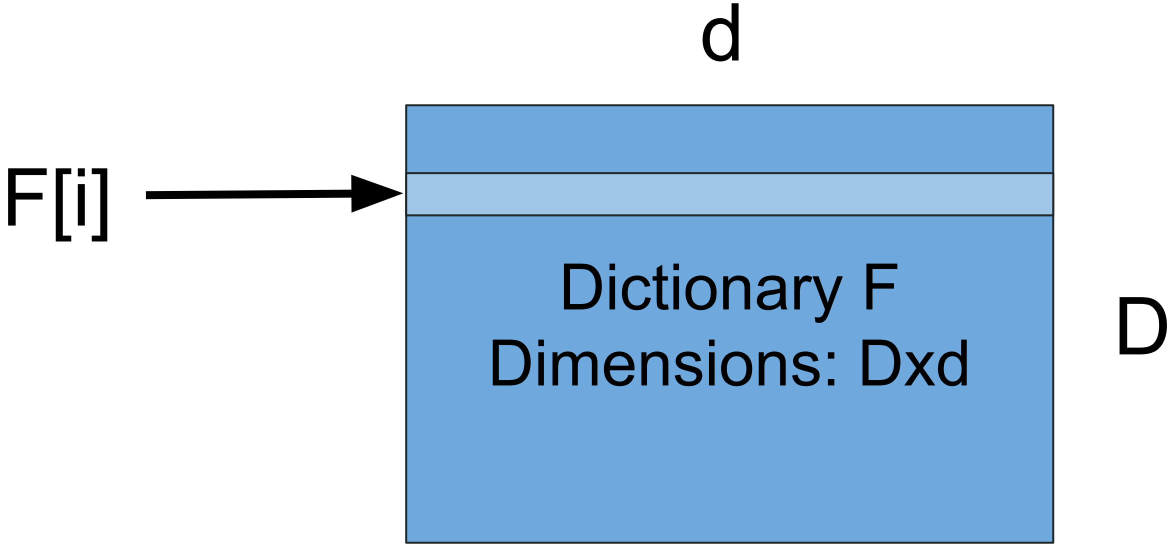
Entities in StarSpace



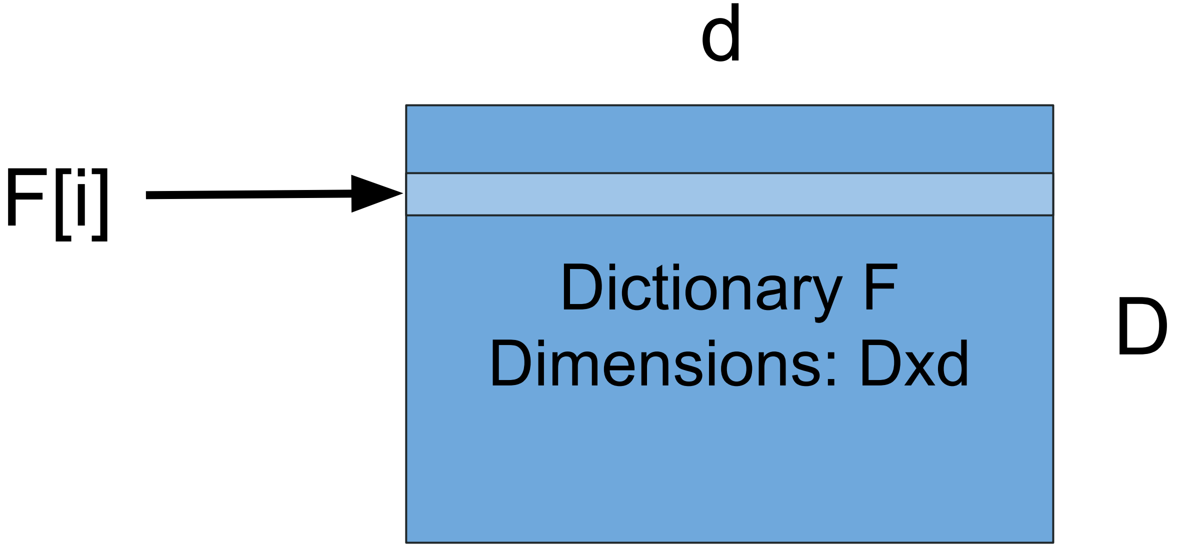
Entities in StarSpace



Entities in StarSpace



Entities in StarSpace



$$\text{Embedding(a)} = \sum_{i \; in \; a} F[i]$$

Entities in StarSpace

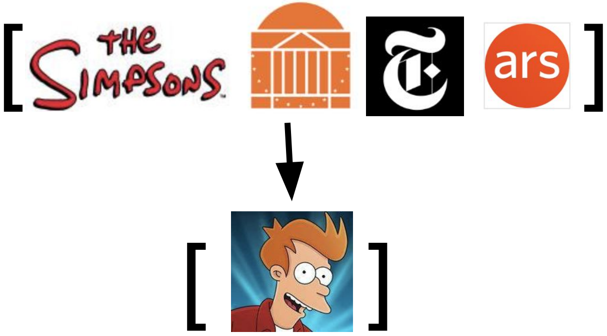
- Embed “Type A” entities and “Type B” entities in the same vector space
    - (a, b) = (document, label)
    - (a, b) = (user, item to recommend)
    - (a, b) = (sentence, sentence)

Entities in StarSpace

- Embed “Type A” entities and “Type B” entities in the same vector space
  - (a, b) = (document, label)
  - (a, b) = (user, item to recommend)
  - (a, b) = (sentence, sentence)
- Document: bag of words
- Label: singleton feature (a word)

Entities in StarSpace

- Embed "Type A" entities and "Type B" entities in the same vector space
  - (a, b) = (document, label)
  - (a, b) = (user, item to recommend)
  - (a, b) = (sentence, sentence)
- Document: bag of words
- Label: singleton feature (a word)
- User: bag of items they've liked
- Item to recommend: single feature (e.g., a Facebook page)



Loss Function

$$\sum_{\substack{(a,b)\in E^{+} \\ b^{-}\in E^{-}}} L^{batch}(sim(a,b), sim(a,b_1^{-}), \ldots, sim(a,b_k^{-}))$$

Loss Function

$$\sum_{\substack{(a,b)\in E^{+} \\ b^{-}\in E^{-}}} L^{batch}(sim(a,b), sim(a,b_1^{-}), \ldots, sim(a,b_k^{-}))$$

Loss Function

$$\sum_{\substack{(a,b)\in E^{+} \\ b^{-}\in E^{-}}} L^{batch}(sim(a,b), sim(a,b_1^{-}),\ldots, sim(a,b_k^{-}))$$

Outline

1. Background: Language Models and Previous Embedding Algos

2. Motivation of StarSpace

3. What is StarSpace? What is new about it?

4. Results

5. Conclusions

Results

- StarSpace vs fastText on Wikipedia dataset

Metric	Hits@1	Hits@10	Hits@20	Mean Rank	Training Time
Unsupervised methods					
TFIDF	24.79%	35.53%	38.25%	2523.68	-
fastText (public Wikipedia model)	5.77%	14.08%	17.79%	2393.38	-
fastText (our dataset)	5.47%	13.54%	17.60%	2363.74	40h
StarSpace (word-level training)	5.89%	16.41%	20.60%	1614.21	45h
Supervised methods					
SVM Ranker BoW features	26.36%	36.48%	39.25%	2368.37	-
SVM Ranker: fastText features (public)	5.81%	12.14%	15.20%	1442.05	-
StarSpace (sentence pair training)	30.07%	50.89%	57.60%	422.00	36h
StarSpace (word+sentence training)	25.54%	45.21%	52.08%	484.27	69h

Results (10 tasks)

- StarSpace word and sentence-level models individually underperformed compared to word2vec and GloVe
    - Word2vec or GloVe had higher accuracy for 8/10 tests

Results (10 tasks)

- StarSpace word and sentence-level models individually underperformed compared to word2vec and GloVe
  - Word2vec or GloVe had higher accuracy for 8/10 tests
- Word + sentence models did better

Results (10 tasks)

- StarSpace word and sentence-level models individually underperformed compared to word2vec and GloVe
  - Word2vec or GloVe had higher accuracy for 8/10 tests
- Word + sentence models did better
- Ensemble word Best accuracy for 4 of the tests+ sentence often even better
  - Best accuracy for 4/10 tests

Outline

1. Background: Language Models and Previous Embedding Algos
2. Motivation of StarSpace
3. What is StarSpace? What is new about it?
4. Results
5. Conclusions

Conclusions

- StarSpace allows greater generality and flexibility
- Succeeds at providing a reasonable baseline for many problems
- Not very efficient--doesn’t use hierarchical classification
- Discrete features, not continuous features