CS378: Natural Language Processing

Lecture 8: Bias in Embeddings, Multilingual Embeddings

Greg Durrett
TEXAS
The University of Texas at Austin

Announcements

- Assignment 2 due in one week
- Bias in embeddings response due next Tuesday (submit on Canvas)
- Nanyun Peng talk Friday; 11am 6.302

Recap

Playing around with embeddings

- See Instapoll

Using Word Embeddings

- Approach 1: learn embeddings as parameters from your data
  - Often works pretty well
- Approach 2: initialize using GloVe, keep fixed
  - Faster because no need to update these parameters
- Approach 3: initialize using GloVe, fine-tune
  - Works best for some tasks

Beyond Word Embeddings

fastText: Sub-word Embeddings

- Same as SGNS, but break words down into n-grams with n = 3 to 6

  where:
  3-grams: <wh, whe, her, ere, re>
  4-grams: <whe, wher, here, ere>,
  5-grams: <wher, where, here>,
  6-grams: <where, where>

- Replace $w \cdot c$ in skip-gram computation with $\left(\sum_{g \in \mathrm{ngrams}} w_g \cdot c\right)$

Bojanowski et al. (2017)

Preview: Subword Tokenization

- Words are a difficult unit to work with, word vocabularies get very large
- Character-level models don't work well
- Compromise solution: use thousands of "word pieces" (which may be full words but may also be parts of words)

Input: _the **_eco tax** _port i co _in _Po nt - de - Bu is …

- Rare words (ecotax, portico, Pont-de-Buis) all get broken up into smaller units we can embed

Sennrich et al. (2016)

Preview: Subword Tokenization

(a)	**Original:**	furiously	(b)	**Original:**	tricycles
	BPE:	_fur | iously		**BPE:**	_t | ric | y | cles
	Unigram LM:	_fur | ious | ly		**Unigram LM:**	_tri | cycle | s
(c)	**Original:**	Completely preposterous suggestions			
	BPE:	_Comple | t | ely | _prep | ost | erous | _suggest | ions			
	Unigram LM:	_Complete | ly | _pre | post | er | ous | _suggestion | s			

- Byte-pair encoding (BPE) produces less linguistically plausible units than another technique based on a unigram language model

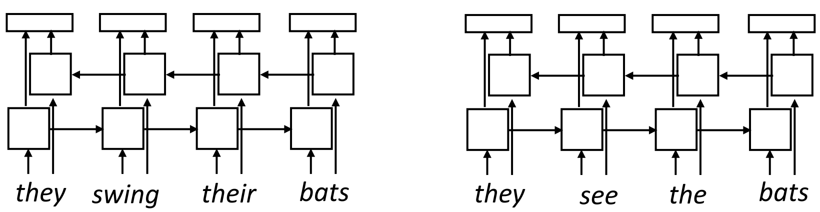
Bostrom and Durrett (2020)

Sentence Embeddings

- What if we want embedding representations for whole sentences?
- Skip-*thought* vectors (Kiros et al., 2015), similar to skip-gram generalized to a sentence level (more later)
- Is there a way we can compose vectors to make sentence representations? Summing?
- Will return to this in a few weeks as we move on to syntax and semantics

Preview: Context-dependent Embeddings

- How to handle different word senses? One vector for *bats*



- ELMo: train a neural language model to predict the next word given previous words in the sentence, use its internal representations as word vectors
- *Context-sensitive* word embeddings: depend on rest of the sentence
- *Huge* improvements across nearly all NLP tasks over GloVe

Peters et al. (2018)

Bias in Word Embeddings

What can go wrong with word embeddings?

- What's wrong with learning a word's "meaning" from its usage? Maybe some words are used in ways we don't want to replicate?
- What data are we learning from?
- What are we going to learn from this data?

Bias Exercise

Answer the following in <=3 sentences each.

Consider learning word embeddings from a **corpus of news articles.**

1. Think about a similarity association a model might learn that you believe constitutes **bias.** For this association, list (a) what the word pair is; (b) why you think this is present in the data (e.g., give an example of how it could appear in a news story)

2. Embeddings are often used at the input layer of a neural network. Can you think of a task for which this biased association might lead to bias in the system?

Now consider learning word embeddings from a **corpus of social media data comments (think about reddit + Twitter).**

3. Do you think you're likely to see the bad association from above? Why or why not?

4. Come up with a new biased similarity association; list (a) what the word pair is; (b) why you think this is present in social media data

What do we mean by bias?

- Compare distance (using cosine similarity) of many occupations to the vectors for *he* and *she*

$$\cos(u,v) = \frac{u \cdot v}{\|u\|\|v\|}$$

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

- These regularities are not restricted to gendered pronouns. *receptionist* is closer to *softball* than *football*
- This work focuses on binary gender stereotypes, but it can be extended

Bolukbasi et al. (2016)

What do we mean by bias?

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

Bolukbasi et al. (2016)

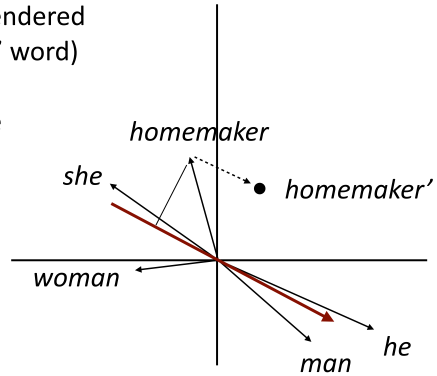
Racial Analogies	
black → homeless	caucasian → servicemen
caucasian → hillbilly	asian → suburban
asian → laborer	black → landowner
Religious Analogies	
jew → greedy	muslim → powerless
christian → familial	muslim → warzone
muslim → uneducated	christian → intellectually

Manzini et al. (2019)

- Nearest neighbor of (b - a + c)

Debiasing

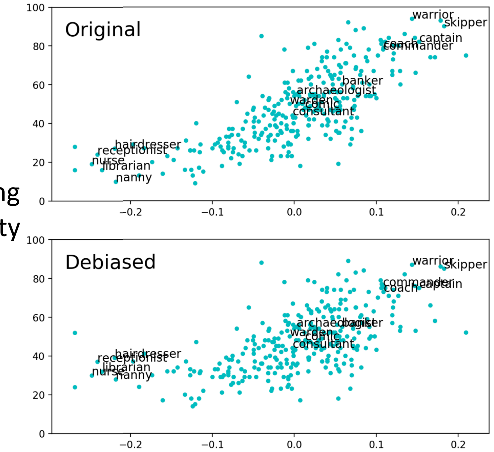
- Identify gender subspace with gendered words (avg “male” - avg “female” word)
- Project words onto this subspace
- Subtract those projections from the original word



Bolukbasi et al. (2016)

Hardness of Debiasing

- Not that effective…and the male and female words are still clustered together
- Bias pervades the word embedding space and isn’t just a local property of a few words



Gonen and Goldberg (2019)

Toxicity

- “Toxic degeneration”: neural models that generate toxic stuff

- System trained on a big chunk of the Internet: conditioning on “SJW”, “black” gives the system a chance of recalling bad stuff from its training data

https://toxicdegeneration.allenai.org/

Multilingual Word Embeddings

Recall: Training Embeddings

- Input: a large corpus of text in some language (English)
- Output: embedding for each word
- What if we have *multiple corpora* of text in *different languages*?
- Learning embeddings on each language individually: these embeddings aren't expected to have any relation

Multilingual Embeddings

- Input: corpora in many languages. Output: embeddings where similar words *in different languages* have similar embeddings

I have an apple
47 24 18 427

J' ai des oranges
47 24 89 1981

ID: 24
ai have

ID: 47
I Je J'

- multiCluster: use bilingual dictionaries to form clusters of words that are translations of one another, replace corpora with cluster IDs, train "monolingual" embeddings over all these corpora
- Works okay but not all that well

Ammar et al. (2016)

Aligning existing embeddings

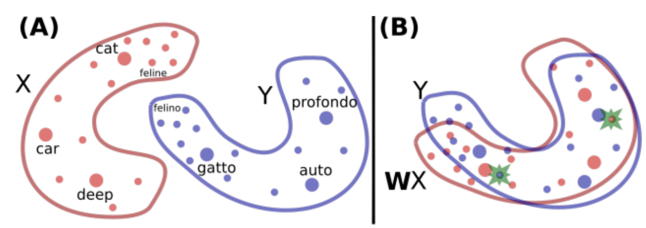
- What if you already have embeddings in two languages and you just want to align them?
- Given: dictionary of pairs (x_i, z_i), where *x* are word embeddings in a source lang (English) and *z* are word embeddings in a target lang (French)
- Learn a matrix *W* to minimize the following:

$$\min_W \sum_{i=1}^{n} \|Wx_i - z_i\|^2$$

(Looks like a loss function! Can learn with SGD on the pairs)

Mikolov et al. (2013)

Aligning existing embeddings



- Rotation learns to align these word embedding spaces! Does this cartoon match reality?

Conneau et al. (2017)

Aligning existing embeddings

Table 2: Accuracy of the word translation methods using the WMT11 datasets. The Edit Distance uses morphological structure of words to find the translation. The Word Co-occurrence technique based on counts uses similarity of contexts in which words appear, which is related to our proposed technique that uses continuous representations of words and a Translation Matrix between two languages.

Translation	**Edit Distance**		**Word Co-occurrence**		**Translation Matrix**		**ED + TM**		**Coverage**
	P@1	P@5	P@1	P@5	P@1	P@5	P@1	P@5	
En → Sp	13%	24%	19%	30%	33%	51%	43%	60%	92.9%
Sp → En	18%	27%	20%	30%	35%	52%	44%	62%	92.9%
En → Cz	5%	9%	9%	17%	27%	47%	29%	50%	90.5%
Cz → En	7%	11%	11%	20%	23%	42%	25%	45%	90.5%

Mikolov et al. (2013)