Better Word Representations with Recursive Neural Networks for Morphology

Topics In Natural Language Processing
23rd February 2016

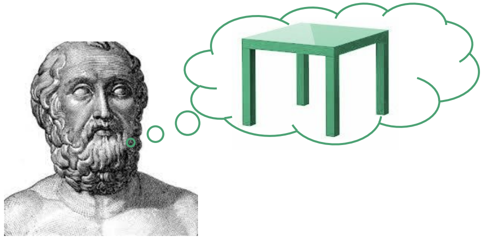
Based on: Luong et al., CoNLL (2013)
Presented by: Paul W. Coles s1523545@sms.ed.ac.uk

Contents

Background

Background: Why is lexical meaning a hard problem? [a ***brief*** view!]

TABLE



= { tables }

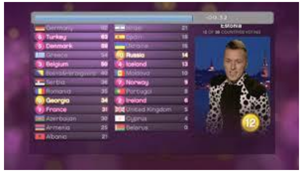
er, what about ‘game’?

greatest integer

HAPPINESS → ?

WITH → ?

Polysemy:



Combinatorial semantics:

```
dining table
camping table
```

Firth, 1957

You shall know a word by the company it keeps.

Background: Vector-Space Lexical Semantics

Occurrence Matrix

	d_1 = IKEA catalogue	d_2 = Wikipedia article 'Earth'	d_3 = climate report	… d_n
w_1 = table	643	12	33	…
w_2= chair	432	0	21	…
w_3= environment	23	54	553	…
… w_n	…	…	…	…

Frequency counts

Other vector spaces possible (e.g. tf-idf).

Based On: Landauer & Dumais (1997)

Background: Vector-Space Lexical Semantics

vector lexical representations

table	environment	chair
122	11	11
133	5	5
75	13	13
444	225	225
92	1	1
14	3	3
6	25	25
3	53	53
...	...	...

cosine angle

(or other vector similarity measure)

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

word similarity

1 => identical

table:chair

chair:environment

0 => orthogonal

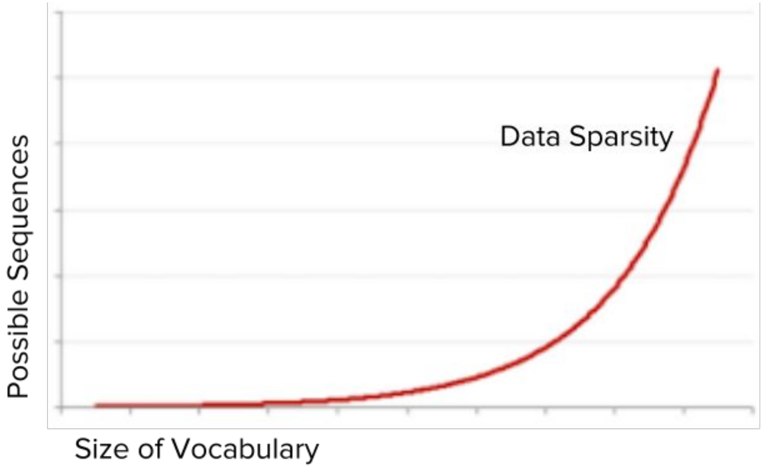
Background: Neural-Net Language Modelling

Old-School Language Modelling:

$$P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{\mathrm{count}(w_{i-(n-1)}, \ldots, w_{i-1}, w_i)}{\mathrm{count}(w_{i-(n-1)}, \ldots, w_{i-1})}$$

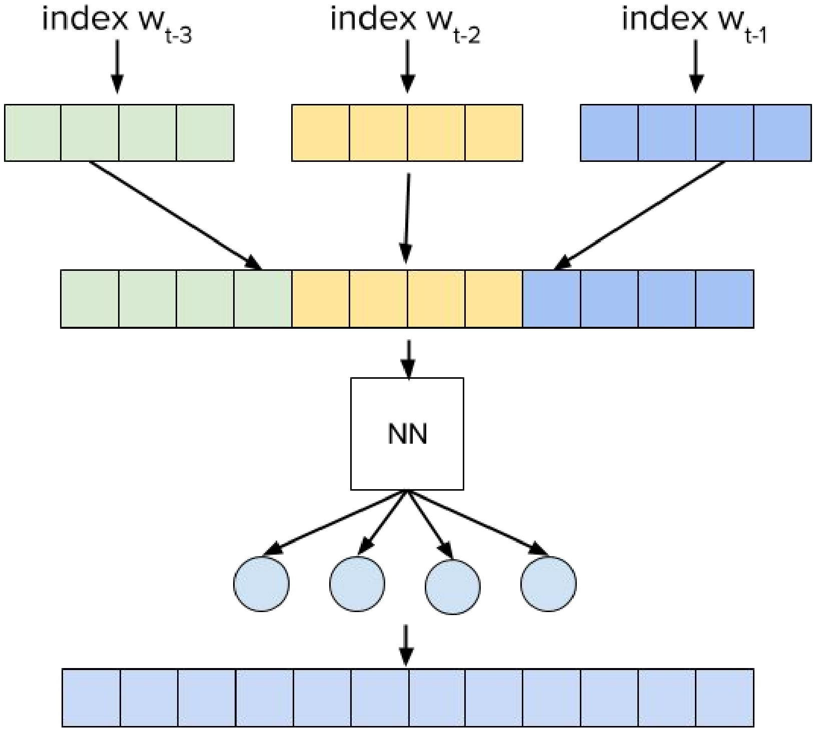
$$p('city' \mid 'Edinburgh\ is\ a') = \frac{C('Edinburgh\ is\ a\ city')}{C('Edinburgh\ is\ a')}$$

Problem: the ‘curse of dimensionality’:



Background: Neural-Net Language Modelling

Neural Language Model (NLM) - conceptual view



1. Look-up embedding for each context word from the matrix $\mathbf{C}$

2. Concatenate to make the neural net input vector $\boldsymbol{X}$

3. Train the net: forward pass, error function and back propogation

4. Apply softmax function to final hidden layer to give conditional distribution over the whole vocabulary: a vector where the $\mathbf{i}^{\mathbf{th}}$ element =

$$P(\, w_t = i \mid \text{context}\,)$$

Background: Neural-Net Language Modelling

Neural Language Model (NLM) - generalises to unseen contexts

‘the man sat down’ not in training data, but ‘the boy sat down’ is

n-gram model (unsmoothed) assigns 0-probability to the ‘the man sat down’:

P(‘down’ | ‘the man sat’) = 0 P(‘down’ | ‘the boy sat’) > 0

assuming ‘boy’ and ‘man’ have similar embeddings, NLM assigns a similar, non-zero probability to both, even if one of these 4-grams is unseen in training

P(‘down’ | ‘the man sat’) > 0 P(‘down’ | ‘the boy sat’) > 0

what about if ‘luckily’, ‘unluckily’ and ‘fortunately’ are in the training data, but ‘unfortunately’ isn’t?

how is this a different case to ‘boy’ vs. ‘man’?

Assumption:

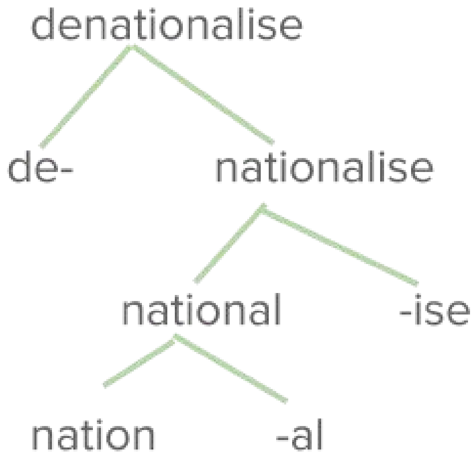
We should represent <u>words</u>

Are words the best linguistic category for capturing semantic distinctions?

Are ‘words’ even a coherent category? Do they even exist?

Background: Natural Language Morphology

Derivation



Inflection

Present Indicative:
bibo
bibes
bibe
bibemos
bibéis
biben

Imperfect:
bibía
bibías
bibía
bibíamos
bibíais
bibían

Preterite:
bibí
bibiste
bibió
bibimos
bibisteis
bibieron

Future:
biberé
biberás
biberá
biberemos
biberéis
biberán

Conditional:
bibería
biberías
bibería
biberíamos
biberíais
biberían

Imperative:
bibe
biba
bibamos
bibed
biban

Present Subjunctive:
biba
bibas
biba
bibamos
bibáis
biban

Imperfect Subjunctive:
bibiera
bibieras
bibiera
bibiéramos
bibierais
bibieran

Gerund:
bibiendo

Past Participle:
bibido

Background: Natural Language Morphology

Productivity: recombination, preservation of meaning, neologism

```
Canaan -> Canannite

Cameron -> Cameronite
```

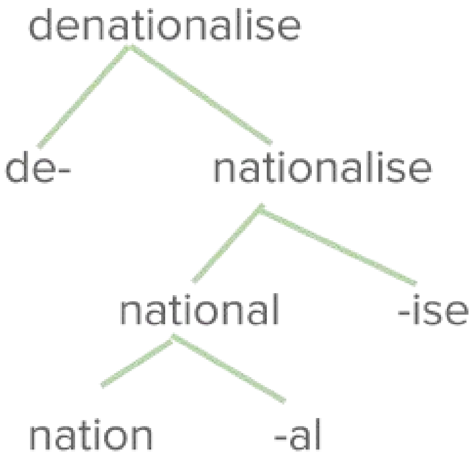
Minimal meaning-bearing unit: the morpheme

```
Cameron

-ite
```

Background: Natural Language Morphology

Just more syntax?



Anglocentrism?

concatenation:

`affix* + stem + suffix*`

fusional language (e.g. Estonian):

`multi-function morphemes`

agglutinative language (e.g. Turkish):

`all-in-one words`

analytic language (e.g. Vietnamese):

`morpheme = word`

Complexity and Frequency

““distinctness” and “unconcerned” are very rare, occurring only 141 and 340 times in Wikipedia documents, even though their corresponding stems “distinct” and “concern” are very frequent (35323 and 26080 respectively).”

Morphologically-complex words occur less frequently

(Zipf Distribution)

But are equally meaningful to speakers!

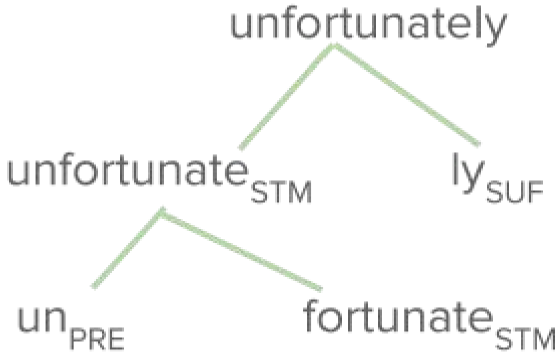
2. Representing Morphology with Recursive Neural Networks

Morphological RNNs: Reference Morphological Representations

‘Gold standard’ for comparison

‘Morfessor’ segmentation toolkit

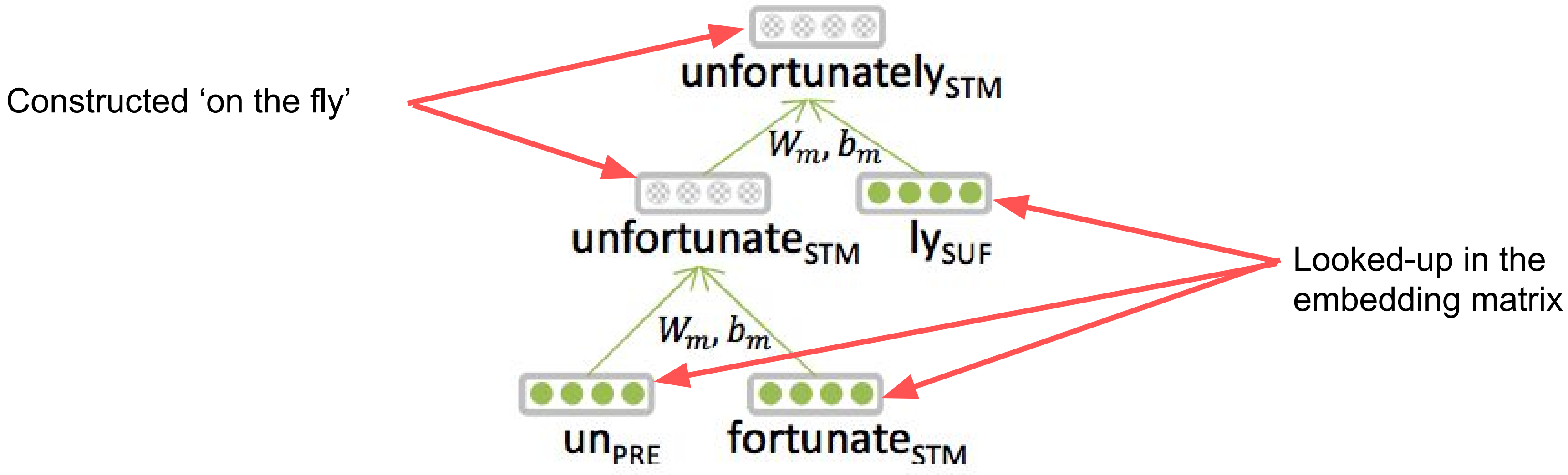
Takes complexes, splits recursively, labels the morphemes:



Result: general word structures like **(pre* stm suf*)**$^{+}$

Morphological RNNs: Context-Insensitive Morphological RNNs

Goal: Construct representations for unseen morphologically-complex words that closely match reference representations



Morphological RNNs: Context-Insensitive Morphological RNNs

Goal: Construct representations for unseen morphologically-complex words that closely match reference representations

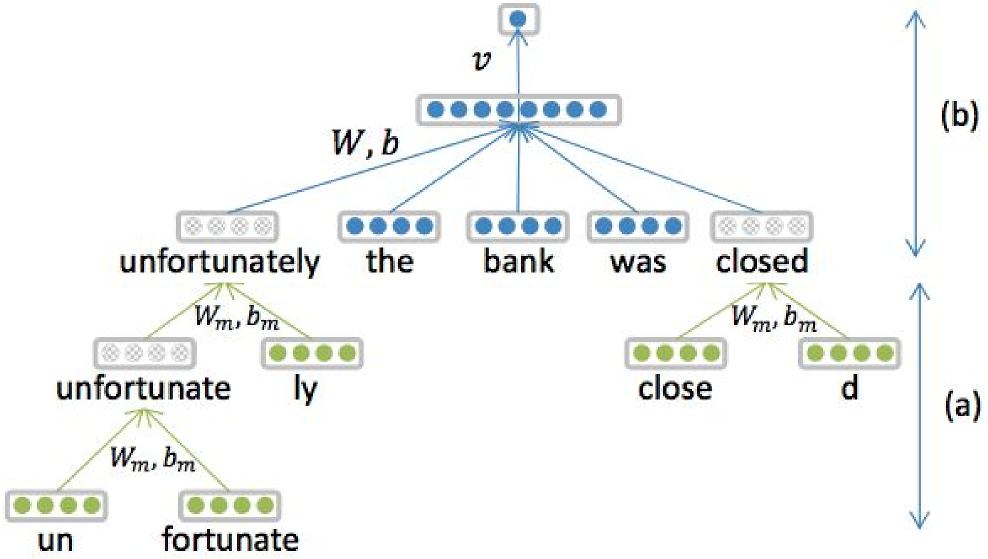
Objective function: how different is the RNN output vector from target?

For each morphological complex x_i in a set of N training examples, define:

Reference Vector:	$p_r(x_i)$	*Computed from 'Morfessor'*
Constructed Vector:	$p_c(x_i)$	*RNN Output*
Cost Function:	$s(x_i) = \| p_c(x_i) - p_r(x_i) \|^2$	*For each training example, x*
Objective Function:	$J(\theta) = \sum_{i=1}^{N} s(x_i) + \frac{\lambda}{2} \|\theta\|_2^2$	*Normalised sum over all examples*

Morphological RNNs: Context-Sensitive Morphological RNNs

Goal: Use NLM training to learn embeddings, but for morphologically-complex words construct representations out of their morphemes



b) word-based neural language model which optimises scores for relevant n-grams

a) the morphological RNN, which constructs representations for words from their morphemes

Morphological RNNs: Context-Sensitive Morphological RNNs

Use NLM to assign a score to each n-gram, $\mathbf{n_i}$, that consists of words $\boldsymbol{x}_1$ to $\boldsymbol{x}_n$:

$$s(n_i) = \boldsymbol{v}^\top f(\boldsymbol{W}[\boldsymbol{x}_1; \ldots; \boldsymbol{x}_n] + \boldsymbol{b})$$

where

$$\boldsymbol{W} \in \mathbb{R}^{h \times nd}$$
$$\boldsymbol{b} \in \mathbb{R}^{h \times 1}$$
$$\boldsymbol{v} \in \mathbb{R}^{h \times 1}$$

Objective function:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{N} \max \begin{cases} 0 \\ 1 - s(n_i) + s(\overline{n}_i) \end{cases}$$

3. Discussion

Evaluation

- Take Wiki snapshot, perform text normalisation
- Candidate pairing from WordNet synsets, human similarity ratings
- 50-dimensional embeddings for words and morphemes based on 10 word windows
- Benchmark performance at word similarity task over standard datasets. These lack morphologically-complex words, so also test on new ‘rare words’ dataset
    - Note: v. rare words perfectly understandable (‘acquirement’)
    - Made using statistics on frequency in Wikipedia
- Compare performance to Collobert et al and Huang et al embeddings
- Conclusion: context-sensitive RNN model outperforms baseline models on all datasets at word-similarity task

Conclusions

- Combining RNN and NLM means “better” word representations are learned
- Two advantages:
    - deals with rare, complex words
    - gives “more principled” way to handle unknown tokens (construct from morphemes)
- They claim:
    - given that English has weak inflectional morphology, the system could “yield even better performance” applied to morphologically-rich languages (Turkish, Finnish)
    - -- they don’t mention non-concatenative languages

Some Questions...

- Isn't this structure implicit in existing word embeddings? My Mikolov-trained model knows what a plural noun is!
    - Mikolov-style embeddings might distinguish 'apple' and 'apples' and extend this to 'table', 'tables'. But '+s for plural' is pretty simple as morphological operations go...
- Actual natural-language morphology vs. stem-affix concatenation operation:
    - Good luck with Hebrew...
    - ...or Mandarin
    - ...or 'be' and 'is'...
- Is word similarity so good an indication of semantic understanding? Any extrinsic examples of this actually helping?
    - QA?
    - IR?
    - MT?

Questions?

Paul W. Coles
s1523545@sms.ed.ac.uk