Training on Lexical Resources

Kenneth Church, Xingyu Cai, Yuchen Bian

Baidu, USA

gft (general fine-tuning):

A Little Language for Deep Nets
(Unix Philosophy: *Less is More*)

Standard 3-Step Recipe

Step	gft Support	Description	Time	Hardware
1		Pre-Training	Days/Weeks	Large GPU Cluster
2	*gft_fit*	Fine-Tuning	Hours/Days	1+ GPUs
3	*gft_predict*	Inference	Seconds/Minutes	0+ GPUs

- Terminology borrowed from *sklearn*:
  - *fit*: $f_{pre} + data \rightarrow f_{post}$
  - *predict*: $f(x) \rightarrow \hat{y}$
- *fit* and *predict* are (almost) all you need
  - *gft* programs are short (1-line)
  - No (not much) programming required
    - No python in this tutorial
    - Examples on hubs are (unnecessarily) long/complicated

Examples of 1-line GFT Programs

Step 2: *gft_fit*

```
gft_fit --eqn 'classify: label ~ text' \
    --model H:bert-base-cased \
    --data H:emotion \
    --output_dir $outdir
```

Data

f_{pre}: Pre-trained Model

f_{post}: Post-trained Model

Step 3: *gft_predict*

```
# text-classification: sentiment analysis
echo 'I love you.' | gft_predict --task text-classification
# I love you.   POSITIVE    0.9998705387115479
```

x $\hat{y}$ Score

:h/ACL2022_deepnets_tutorial 3

Methods

- Baselines
  - MoE: Mixture of Experts
    - Nguyen et al (2017)
    - with default settings
  - MoE with DLCE embeddings
    - Nguyen et al (2017)
    - with better settings
- Proposed Method
  - gft_fit with f_{pre} = bert (uncased)
  - gft_predict

Datasets

Dataset	train	val	test
adj	5562	398	1986
noun	2836	206	1020
verb	2534	182	908
fallows	58,494	7190	7366
fallows-s	5886	753	777

Fallows (1898)

Table 3: Sizes (edges) of synonym-antonym datasets

Evidence for Leakage

Paths of Length 1

- Consider 99 edges of length 1
  - Example: *good* ↔ *awful*
- These are particularly worrisome.
  - The same edge is in
  - both train and validation splits,
  - but in different directions
- These 99 pairs are clearly leaking information across splits

Many Short Paths

Path Length	adj	noun	verb	fallows
0				2
1	99	59	60	946
2	80	7	15	3835
3	59	3	7	1156
4+	70	2	35	639
NA	90	135	65	612
total	398	206	182	7190

Table 10: For most pairs of words in the validation set, w_1 and w_2, there is a short path from w_1 to w_2 based on edges in the training set.

Syn/Ant Classification → VAD Regression

- Motivation: classify → regress
  - Concerns about leakage
- NRC-VAD is similar to sym/lex
  - but lexicon is fully-connected
- 20k lemmas, w, where $VAD(w) \in \mathbb{R}^3$
  - $y(w_1, w_2) = |VAD(w_1) - VAD(w_2)|$
- Regression: $y \sim w_1 + w_2$
- Standard test/val/train splits:
  - Split lexicon by E
- But for generalizations to OOVs
  - Might be more interested in splits by $V(w)$

word	Val	Arousal	Dom	Dist
open	0.620	0.480	0.569	0.00
unfold	0.612	0.510	0.520	0.06
reopen	0.656	0.528	0.568	0.06
close	0.292	0.260	0.263	0.50
closed	0.240	0.164	0.318	0.55
undecided	0.286	0.433	0.127	0.56

Table 12: Words above the double line are near *open*. The last column is the Euclidean distance to *open*.

	Antonyms				Sim
	adj	noun	verb	fallows	SimLex
cor	0.55	0.48	0.44	0.52	-0.40

Table 13: VAD distances are positively correlated with antonyms, and negatively correlated with SimLex similarities, though none of these correlations are large.

Agenda

- Syn/Ant Binary Classification
- From Words to Texts
  - MWEs: Multiword Expressions
  - OOVs: Out of Vocabulary words
  - Multi-Lingual
  - Negation
- Leakage with Standard Benchmarks
- VAD Regression
  - VAD = Valance, Arousal, Dominance

MoE with better settings

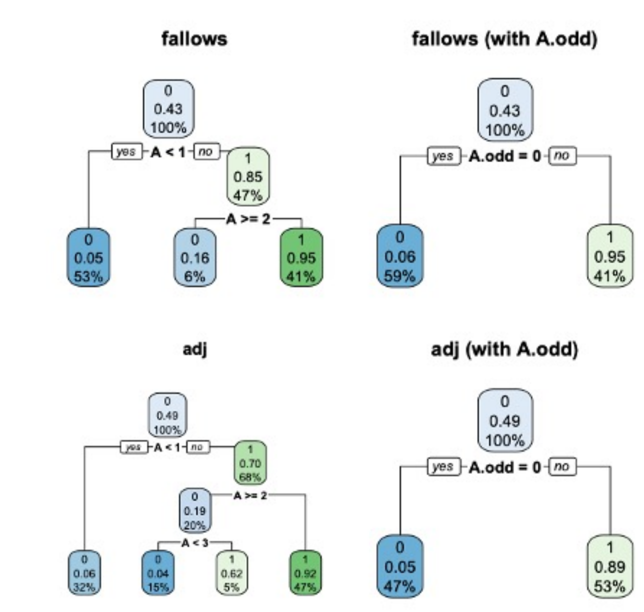
Test	Train				
	adj	noun	verb	fallows	fallows-s
adj	**0.921**	0.859	0.852	0.897	0.868
noun	0.841	**0.917**	0.857	0.828	0.785
verb	0.813	0.829	**0.903**	0.851	0.794
fallow	0.633	0.604	0.620	**0.666**	0.634
fallow-s	**0.659**	0.602	0.591	**0.659**	0.627

Proposed: Fine-Tuning

Test	Train			
	adj	noun	verb	fallows
adj	**0.908**	0.657	0.713	0.881
noun	0.773	**0.877**	0.792	0.797
verb	0.767	0.722	**0.906**	0.867
fallows	0.722	0.610	0.698	**0.947**

A-Leakage

- Definitions
  - Let $e = (a, b)$ be an edge in val split
  - Let $label_v(e)$ be the label on e in val
  - Let $path_t(a, b)$ be the shortest path
    - from a to b using edges from train
  - Let A_t be the number of antonym labels on $path_t(e)$
- *A*-Leakage Heuristic:
  - $label_v(e) \approx$ antonym iff A_t is odd



VAD Results (R2)

R2 → 1.0 (good); R2 → 0 (bad)

- Train/Val/Test splits
  - Based on $V = 16k$ (of $20k$)
  - Remainder held-out to test generalizations to OOVs
- Results are promising when
  - Splits are large and
  - Representative of one another
- Experimented with training sets of 10k, 100k and 1M edges

Promising Transfer: Train with 1M Edges

$R2(test) \approx R2(val) \approx R2(train) \approx 1$

base model	train	val	test
BERTun	0.993	0.993	0.993
SciBERTun	0.993	0.993	0.992
ERNIE	0.991	0.990	0.990
SciBERTc	0.988	0.988	0.987
BERTmulti	0.988	0.987	0.991
BERTc	0.995	0.995	0.988

Training on Fallows Thesaurus

Training (fit)

classify: gold ~ word1 + word2

0 → Synonym
1 → Antonym

word1	word2	gold
ancient	oldfashioned	0
blame	disapprove	0
clearly	confusedly	1
debt	liability	0
demure	modest	0
profitable	fruitless	1
revelry	orgies	0
rotation	order	0
vanity	selfdistrust	1

$y \sim text_1 + text_2$

Inference (predict)

$text_1$	$text_2$	y_1	y_2
good	bad	-3.95	4.54
bad	evil	4.44	-5.00
good	benevolent	4.43	-5.05
bad	benevolent	-3.44	4.16
good	terrorist	-3.43	4.10
bad	terrorist	4.48	-5.10

Table 1: Inference: synonymy iff $y_1 > y_2$

Data from Fallows (1898)

LREC-2022 https://github.com/kwchurch/syn_ant 6

Delta (Proposed – MoE)

Test	Train			
	adj	noun	verb	fallows
adj	-0.013	-0.202	-0.139	-0.016
noun	-0.068	-0.040	-0.065	-0.031
verb	-0.046	-0.107	0.003	0.016
fallows	0.089	0.006	0.078	0.281

Conclusions

- Proposed fine-tuning deep nets on lexical resources
  - Thesaurus (syn/ant classification)
  - VAD Regression
  - $y \sim text_1 + text_2$
- Proposed method is competitive with MoE baseline, and
  - Generalizes better to Fallows (1898)
- Words → Texts
  - Proposed method can be applied at inference time to MWEs, OOVs and longer tests in multiple languages
- On a cautionary note,
  - found evidence of leakage
    - in standard benchmarks as well as Fallows (1898)
  - Work based on bad benchmarks
    - may need to be retracted
- To address concerns with leakage,
  - we introduced a new task: VAD regression
  - Since VAD is fully-connected,
    - we could study sampling methods
- Transfer is more effective
  - when splits are large
    - and representative of one another
  - In such cases,
    - reduces training loss (in fine-tuning)
    - also reduces loss on other splits
- Proposed method:
  - effective for pairs of words in training set
  - but less so for pairs of unseen words

From Words to Text (Undesirable Biases)

Logits (top); Cor of Logits (bottom)

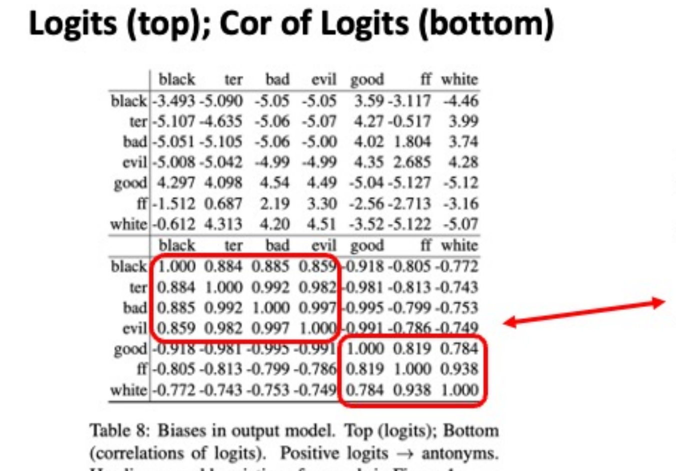
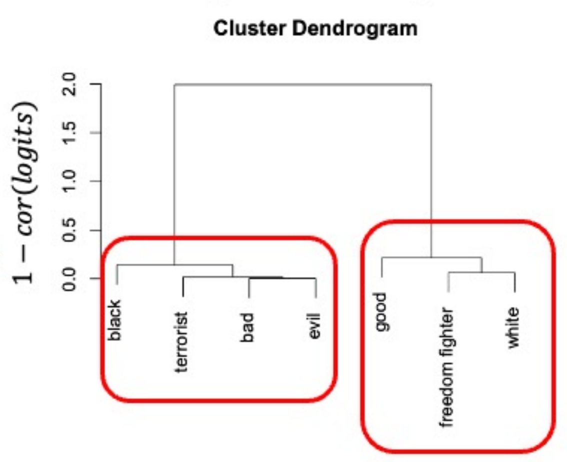


Table 8: Biases in output model. Top (logits); Bottom (correlations of logits). Positive logits → antonyms. Headings are abbreviations for words in Figure 1.

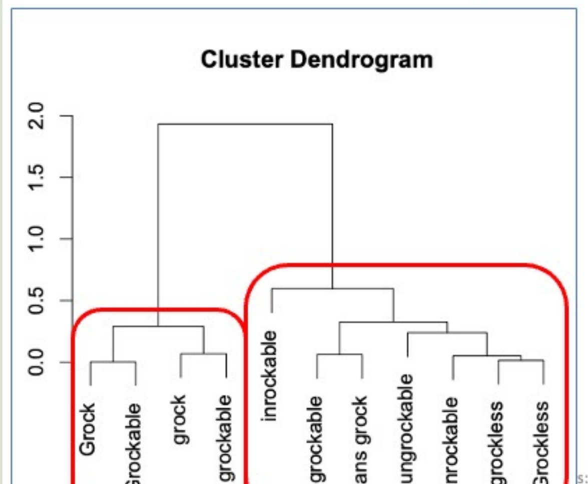
Clustering of Cor of Logits

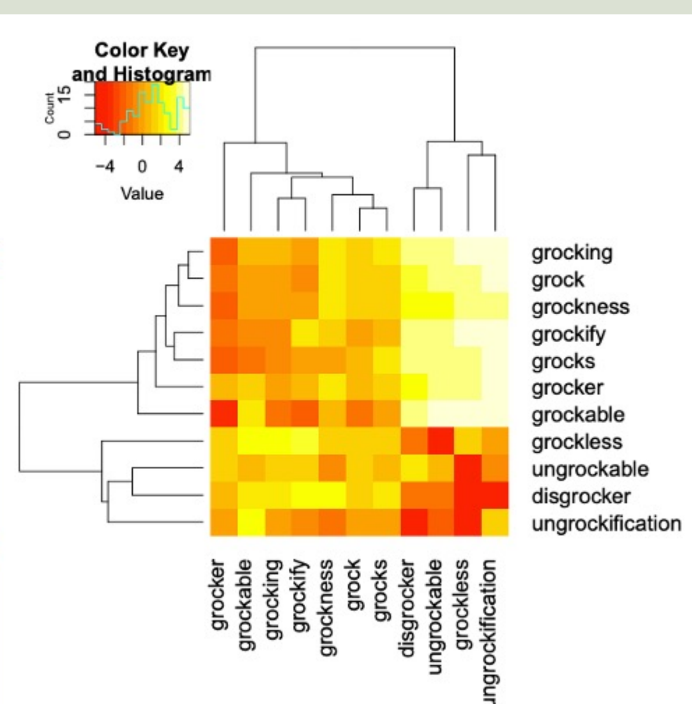


From Words to Text

(OOVs: Out of Vocabulary)

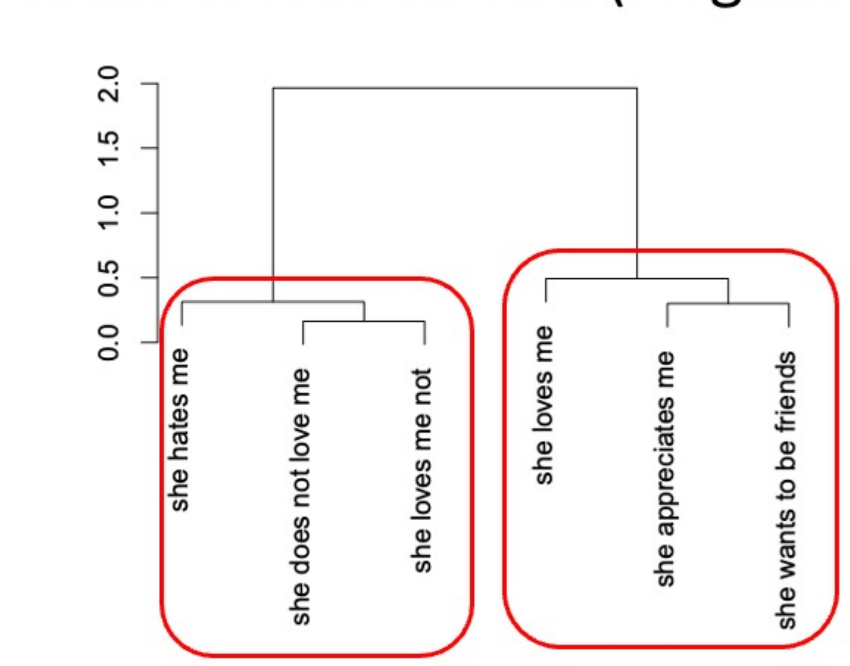
Google Translations of Variants of *Grock*





://github.com/kwchurch/syn_ant 15

From Words to Text (Negation)



Morpheme Diagnostic

- Group words by affixes
  - *over-*
    - *overtake/take*
    - *overlook/look*
- Plot y for pairs in each group
  - $y(w_1, w_2) = |VAD(w_1) - VAD(w_2)|$
- Red baselines:
  - 0: distance for maximally similar pair
  - $\sqrt{2}$: distance for random pair
- Observations:
  - VAD varies systematically:
    - Small (similar in VAD): *-s, -ism, -ly*
    - Large (dissimilar in VAD): *-less, dis-, un-*
  - Word2vec is large (almost everywhere)
    - Almost all pairs of words are far apart
    - Even words that are morphologically related