Lecture 14
Sequence tagging, and social media NLP

Intro to NLP, CS585, Fall 2014

http://people.cs.umass.edu/~brenocon/inlp2014/

Brendan O'Connor

- Added after lecture, will review on Thursday:
  - The three main structpred models are: (1) struct perceptron, (2) crf's, (3) structsvm's. All of them work the same at test time (decoding via the viterbi algorithm, by maximizing a linear goodness score). Only at training time are they different.
  - Averaged perceptron is probably the simplest to implement and use. Lots of practitioners in NLP who don't care about fancy machine learning often use it. I actually like CRF's myself because of they have a probabilistic interpretation, but that doesn't always matter. Training CRF's is slightly more complicated than struct perceptrons (not that much more complicated, but like a lecture's worth of material), so I figured we could skip it in this class.
  - Instead of averaging, you can also do early stopping: keep a development set and evaluate accuracy on it every iteration through the data. Choose the theta that did best. I don't know which method is better (different researchers may prefer different methods). Averaging has the advantage that there aren't really any hyperparameters to tune (well, the learning rate to a certain extent).
  - Why does averaging work? Theta is bouncing a lot around the space, because the perceptron doesn't know how to prefer solutions according to the magnitude of the errors it makes. The value of theta will be overfitted towards doing well on the most recent examples it's seen. If you average, you average away some of the noise. Averaging is used in other areas of machine learning too. It's a form of regularization.
  - Perceptron learning is actually a form of gradient descent. It's not on the logistic regression log-likelihood, but instead the gradients of a different function (the "1-0" loss).
    - The Collins 2002 paper that introduced the structured perceptron is still great to read for more details: http://www.cs.columbia.edu/~mcollins/papers/tagperc.pdf
  - More on the classification perceptron: see Hal Daume's book chapter draft, http://ciml.info/dl/v0_9/ciml-v0_9-ch03.pdf

- Is perceptron learning a form of gradient descent? Yes!
- *Stochastic gradient descent (ascent)* algorithm: on every training example, increment gradient

$$\theta := \theta + \eta \underbrace{g_i(x_i, y_i)}_{\textit{gradient for just one example}}$$

- Log-linear gradient for SGD

log-likelihood:
how good model is at predicting gold y_i

$$g_i = \frac{\partial}{\partial \theta_j} \left[\overline{\theta^\mathsf{T} f(x_i, y_i) - \log \sum_{y' \in \mathcal{Y}} \exp \theta^\mathsf{T} f(x_i, y')} \right]$$

$$= f_j(x_i, y_i) - \underbrace{\sum_{y'} p(y'|x) f_j(x_i, y')}$$

feature's expected value,
under model's prediction distribution

- Perceptron as SGD

$$y^* = \arg\max_y \theta^\mathsf{T} f(x, y)$$

perceptron neg-loss:
different measure of evaluating model predictions

$$g_i = \frac{\partial}{\partial \theta_j} \left[\overline{\theta^\mathsf{T} f(x_i, y_i) - \theta^\mathsf{T} f(x_i, y^*)} \right]$$

$$= f_j(x_i, y_i) - f_j(x_i, y^*)$$

In both cases: loss gradients want to make gold-standard features match predicted-structure features. Either you care about a distribution over all outputs ... or just the best output

Conditional Random Fields
Lafferty, McCallum, Pereira 2001



- Log-linear gradient for SGD

log-likelihood:
how good model is at predicting gold y_i

$$g_i = \frac{\partial}{\partial \theta_j} \left[\theta^\mathsf{T} f(x_i, y_i) - \log \sum_{y' \in \mathcal{Y}} \exp \theta^\mathsf{T} f(x_i, y') \right]$$

$$= f_j(x_i, y_i) - \sum_{y'} p(y'|x) f_j(x_i, y')$$

feature's expected value,
under model's prediction distribution

Structured Perceptron
Collins 2002

- Perceptron as SGD

$$y^* = \arg\max_y \theta^\mathsf{T} f(x, y)$$

perceptron neg-loss:
different measure of evaluating model predictions

$$g_i = \frac{\partial}{\partial \theta_j} \left[\theta^\mathsf{T} f(x_i, y_i) - \theta^\mathsf{T} f(x_i, y^*) \right]$$

$$= f_j(x_i, y_i) - f_j(x_i, y^*)$$

In both cases: loss gradients want to make gold-standard features match predicted-structure features. Either you care about a distribution over all outputs ... or just the best output

approach	$loss(\boldsymbol{x}, \boldsymbol{y}; h)$	training expense	notes
generative models (3.3)	$-\log p_{\mathbf{w}}(\boldsymbol{x}, \boldsymbol{y})$	if multinomial-based, easy to train	can answer "many questions," but the model must explain all evidence
globally normalized conditional models (3.5)	$-\log p_{\mathbf{w}}(\boldsymbol{y} \mid \boldsymbol{x}) = -\mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) + z_{\mathbf{w}}(\boldsymbol{x})$	require inference for feature expectations and $z_{\mathbf{w}}$	allow arbitrary local features; hybridize generative and discriminative approaches
perceptron (3.6.2)	$-\mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) + \max_{\boldsymbol{y}' \in \mathcal{Y}_{\boldsymbol{x}}} \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}')$	only requires a decoder	no probabilistic interpretation or explicit regularization
large margin models (3.6)	$-\mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}) + \max_{\boldsymbol{y}' \in \mathcal{Y}_{\boldsymbol{x}}} \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y}') + cost(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{y})$	only require a cost-augmented decoder	incorporate cost function; no probabilistic interpretation

Figure 3.3: A comparison of the main learning methods discussed in this chapter. The form of the predictor $h(\boldsymbol{x})$ is assumed to be a linear decoder, $\operatorname{argmax}_{\boldsymbol{y} \in \mathcal{Y}_{\boldsymbol{x}}} \mathbf{w}^\top \mathbf{g}(\boldsymbol{x}, \boldsymbol{y})$.

Noah Smith,
Linguistic Structure Prediction, page 107
link on course webpage

Averaging vs. early stopping

- Why does the perceptron keep flip flopping?
- This induces overfitting: cares too much about whatever it last saw
- Solution #1: early stopping
- Solution #2: averaging (or voting...)
  - Averaging seems to be the most popular: no fiddly hyperparameters to tune.
  - Perceptrons don't allow a regularization term ... averaging is an alternate form of anti-overfitting control
- Avg. perceptron seems to be the most popular supervised struct. pred. algorithm for people who don't care about machine learning and just want to do NLP. ("code to usefulness ratio"...)

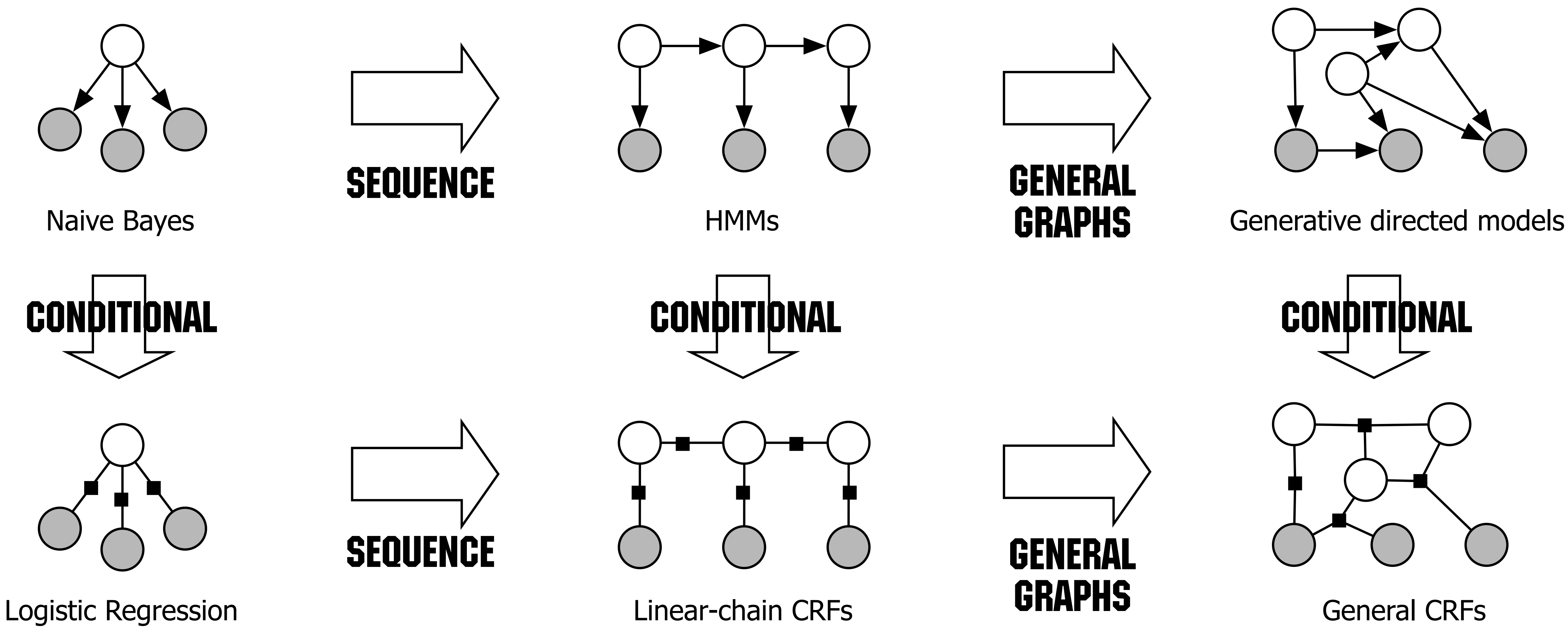


Figure 1.2 Diagram of the relationship between naive Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

From Sutton and McCallum tutorial on CRFs

Applications of sequence tagging

Document segmentation

38 files belonging to 7 UseNet FAQs
Tagging decisions are at the *line* level

Example:

```
<head>                     X-NNTP-Poster: NewsHound v1.33
<head>                     Archive-name: acorn/faq/part2
<head>                     Frequency: monthly
<head>
<question>2.6) What configuration of serial cable should I use?
<answer>
<answer>                       Here follows a diagram of the necessary connection
<answer>                   programs to work properly.  They are as far as I know
<answer>                   agreed upon by commercial comms software developers fo
<answer>
<answer>                      Pins 1, 4, and 8 must be connected together inside
<answer>                   is to avoid the well known serial port chip bugs.  The
```

Features in Experiments

begins-with-number
begins-with-ordinal
begins-with-punctuation
begins-with-question-word
begins-with-subject
blank
contains-alphanum
contains-bracketed-number
contains-http
contains-non-space
contains-number
contains-pipe
contains-question-mark
contains-question-word
ends-with-question-mark
first-alpha-is-capitalized
indented
indented-1-to-4
indented-5-to-10
more-than-one-third-space
only-punctuation
prev-is-blank
prev-begins-with-ordinal
shorter-than-30

Results for FAQ segmentation

Learner	*Segmentation precision*	*Segmentation recall*
Log. Reg.	0.038	0.362
TokenHMM	0.276	0.140
FeatureHMM	0.413	0.529
MEMM	0.867	0.681

Context important!

Features important!

Named Entity Recognition

The task is usually defined as:
identify segments in text that are names,
and some coarse types for them

SOCCER - [PER BLINKER] BAN LIFTED .
[LOC LONDON] 1996-12-06 [MISC Dutch] forward [PER Reggie Blinker] had his indefinite suspension lifted by [ORG FIFA] on Friday and was set to make his [ORG Sheffield Wednesday] comeback against [ORG Liverpool] on Saturday . [PER Blinker] missed his club's last two games after [ORG FIFA] slapped a worldwide ban on him for appearing to sign contracts for both [ORG Wednesday] and [ORG Udinese] while he was playing for [ORG Feyenoord].

Figure 1: Example illustrating challenges in NER.

Example from Ratinov and Roth 2009

BIO-notation

Named Entity Recognition as sequence tagging

PER
LOC
ORG

After meeting with the Denver Post Editorial Board,
O O O O B-ORG I-ORG I-ORG I-ORG

Virginia Lake traveled to Gile State Forest New Hampshire
B-PER I-PER O O B-LOC I-LOC I-LOC B-LOC I-LOC

where she went camping with her daughter Anne's Girl Scout Troop.
O O O O O O O B-PER B-ORG I-ORG I-ORG

- State-of-the-art performance reported in the range 85 to 94% F-score (avg of prec/rec), depending on the annotated dataset
  http://www.aclweb.org/aclwiki/index.php?title=Named_Entity_Recognition_(State_of_the_art)
- But keep in mind your training data and features...
  http://nlp.stanford.edu:8080/corenlp/process
  Who is seeing Skrillex today?

Application:
Social media NLP

- Sequence models for online conversational text
- Why is online conversational text interesting or hard?

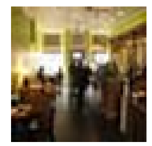
  - Some material borrowed from Jacob Eisenstein
  - Useful resource: Alan Ritter's social media NLP course http://aritter.github.io/courses/5539.html

A partial taxonomy of Twitter messages

Official announcements

BritishMonarchy TheBritishMonarchy
On 6 Jan: Changing the Guard at Buckingham Palace - Starts at approx 11am http://www.royal.gov.uk/G
17 hours ago

Business advertising

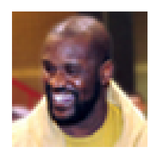


bigdogcoffee bigdogcoffee
Back to normal hours beginning tomorrow........Monday-Friday 6am-10pm Sat/Sun 7:30am-10pm
2 Jan

Links to blog and web content

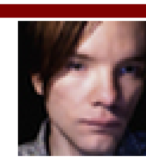
crampell Catherine Rampell
Casey B. Mulligan: Assessing the Housing Sector - http://nyti.ms/hcUKK9
10 hours ago

Celebrity self-promotion



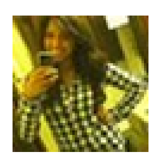
THE_REAL_SHAQ THE_REAL_SHAQ
fill in da blank, my new years shaqalution is ___________
4 Jan

Status messages



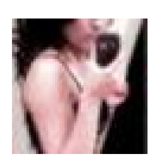
emax electronic max
1.1.11 - britons and americans can agree on the date for once. happy binary day!
1 Jan

Group conversation



_siddx3 Evelyn Santana
RT @_LusciousVee: **#EveryoneShouldKnow** Ima Finally Be 18 This Year ^.^
3 minutes ago

Personal conversation



xoxoJuicyCee CeeCee'♥
@fxknnCelly aha kayy goodnightt (:
4 Jan

Isn’t this “bad language”?

- Text in computer-mediated communication (SMS, social media, IRC....)
  has shortenings, abbreviations, and grammar that’s very different than standard written English.
- Is it “bad language”?
- Why is it so different?

Are users illiterate?

- ... No.

Length limits?

- ... No.

standard	length	alternative	length
your	85.1 ± 0.4	*ur*	81.9 ± 0.6
you're	90.0 ± 0.1		
with	87.9 ± 0.3	*wit*	78.8 ± 0.7
going	82.7 ± 0.5	*goin*	72.2 ± 1.0
know	86.1 ± 0.4	*kno*	78.4 ± 1.0
about	88.9 ± 0.4	*bout*	74.5 ± 0.7

Table 1: Average length of messages containing standard forms and their shortenings

What do you see in conversations?

- Language use is socially contingent
  - Individual (every person has a dialect?)
  - Social groups
  - Gender
  - Socioeconomic background
  - Ethnicity
  - Geographic region....

Minority dialects/languages

- “A language is a dialect with an army and navy”
- Are minority languages/dialects “incorrect”?
  What accomodations are given to minority languages?
  - Ukrainian vs. Russian ...
  - African American dialects vs standard American English ...
- Descriptive linguistics vs. prescriptive grammarians


Create account Log in

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article

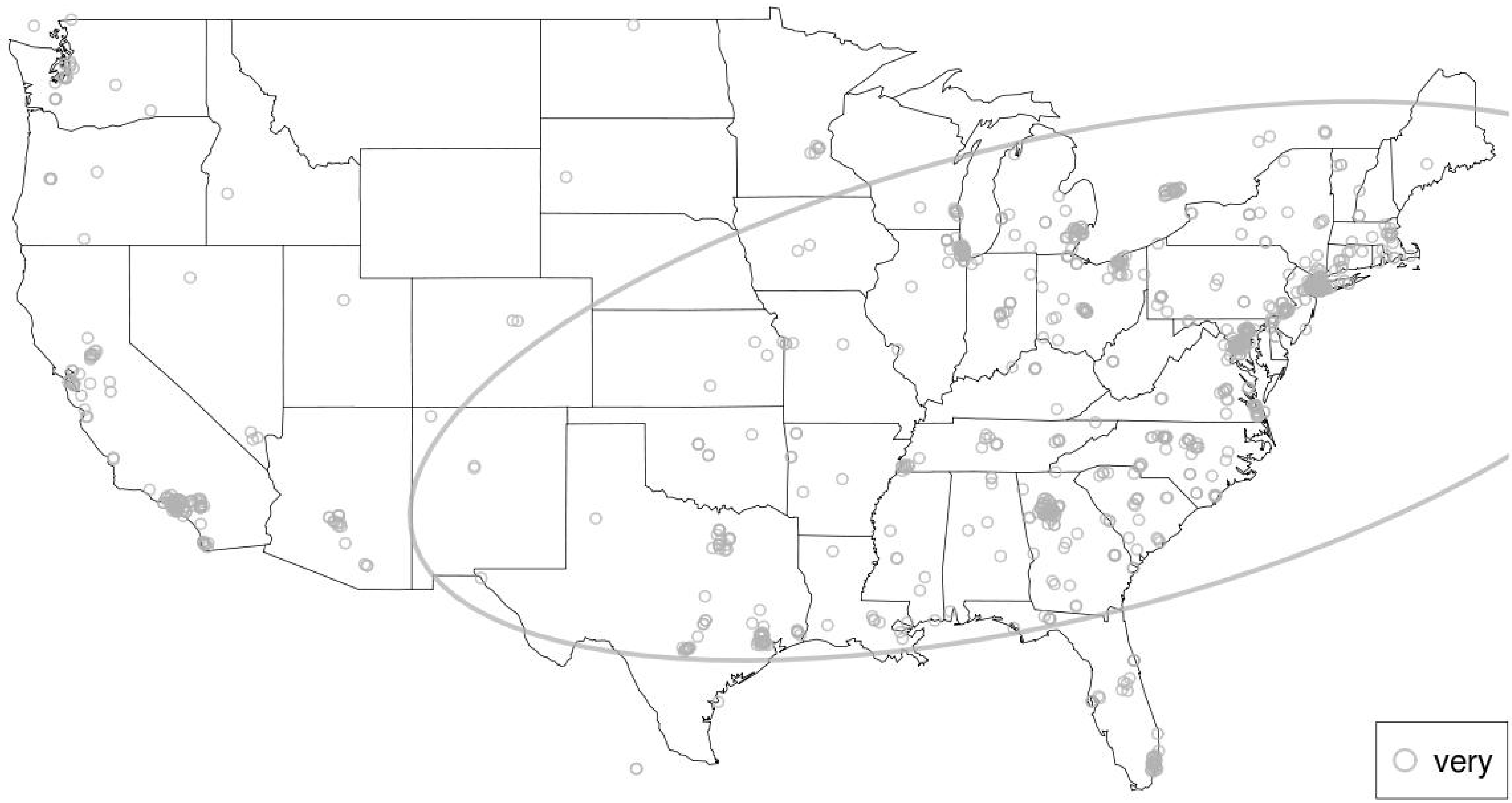
Article Talk Read Edit View history Search

Oakland Ebonics controversy

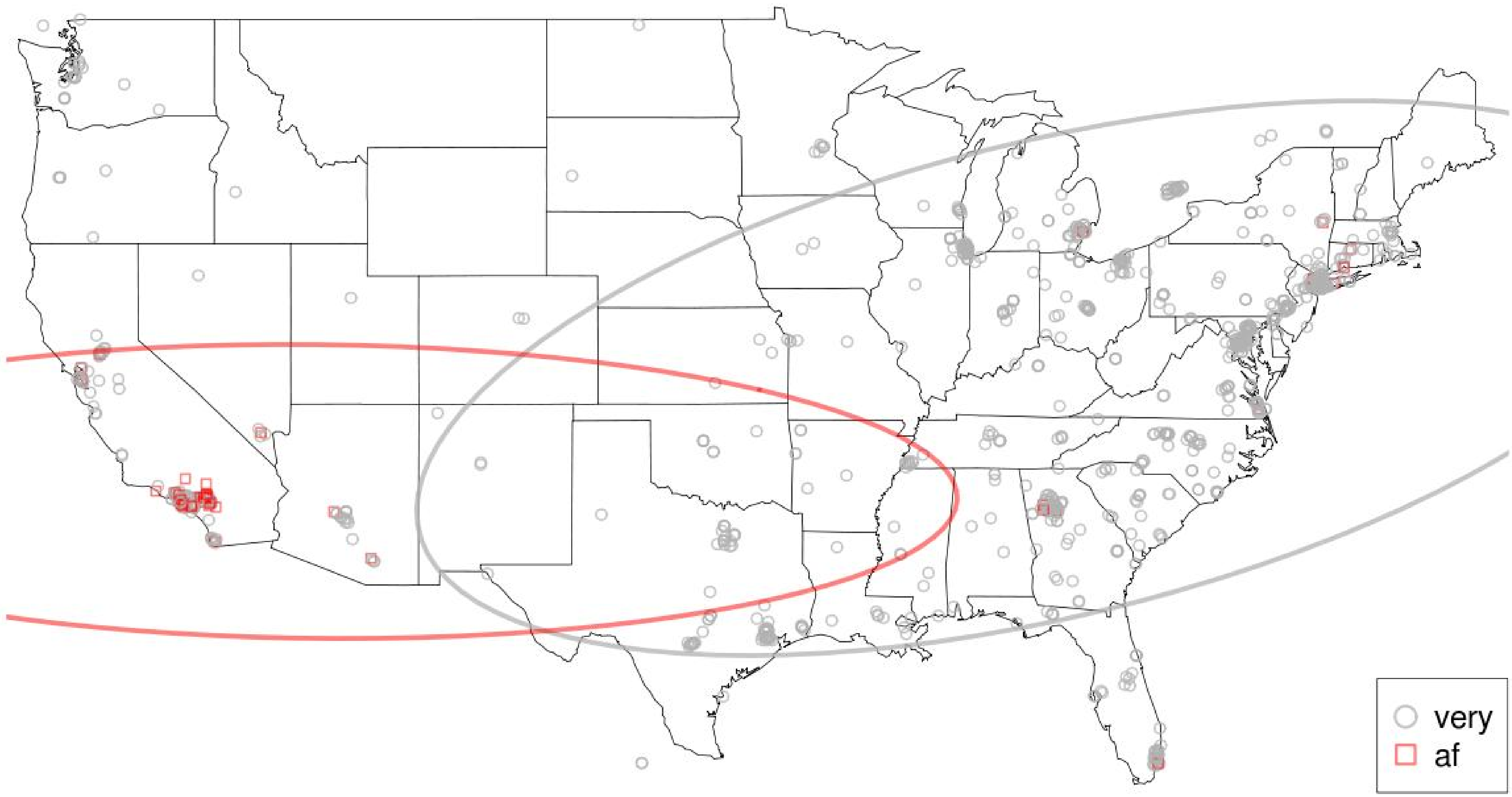
From Wikipedia, the free encyclopedia

On December 18, 1996, the Oakland, California school board passed a controversial resolution recognizing the legitimacy of "Ebonics"—what mainstream linguists more commonly term African-American Vernacular English— (AAVE) as a language. The resolution set off a firestorm of media criticism and ignited a hotly discussed national debate.

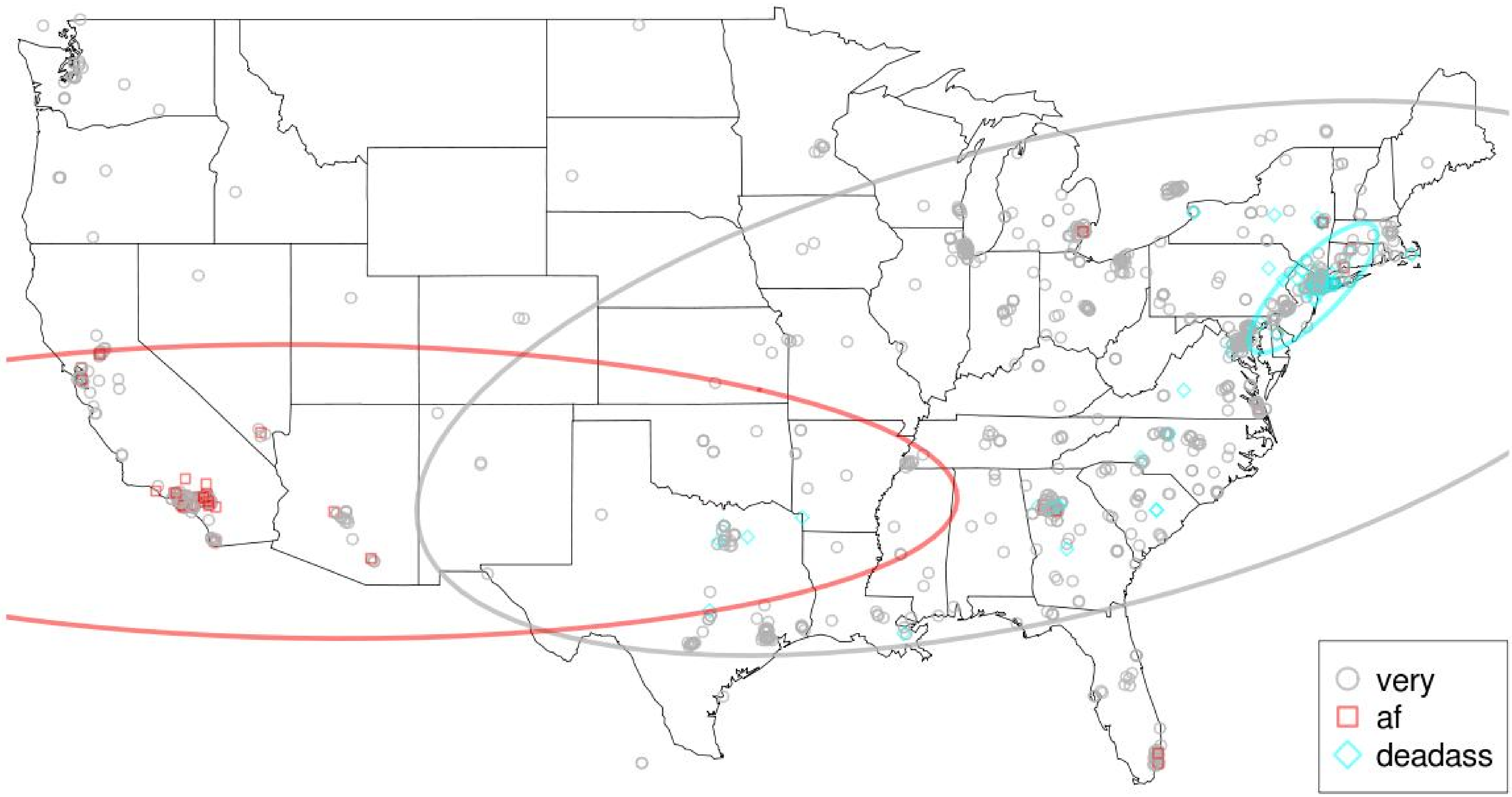
Social contingency of language



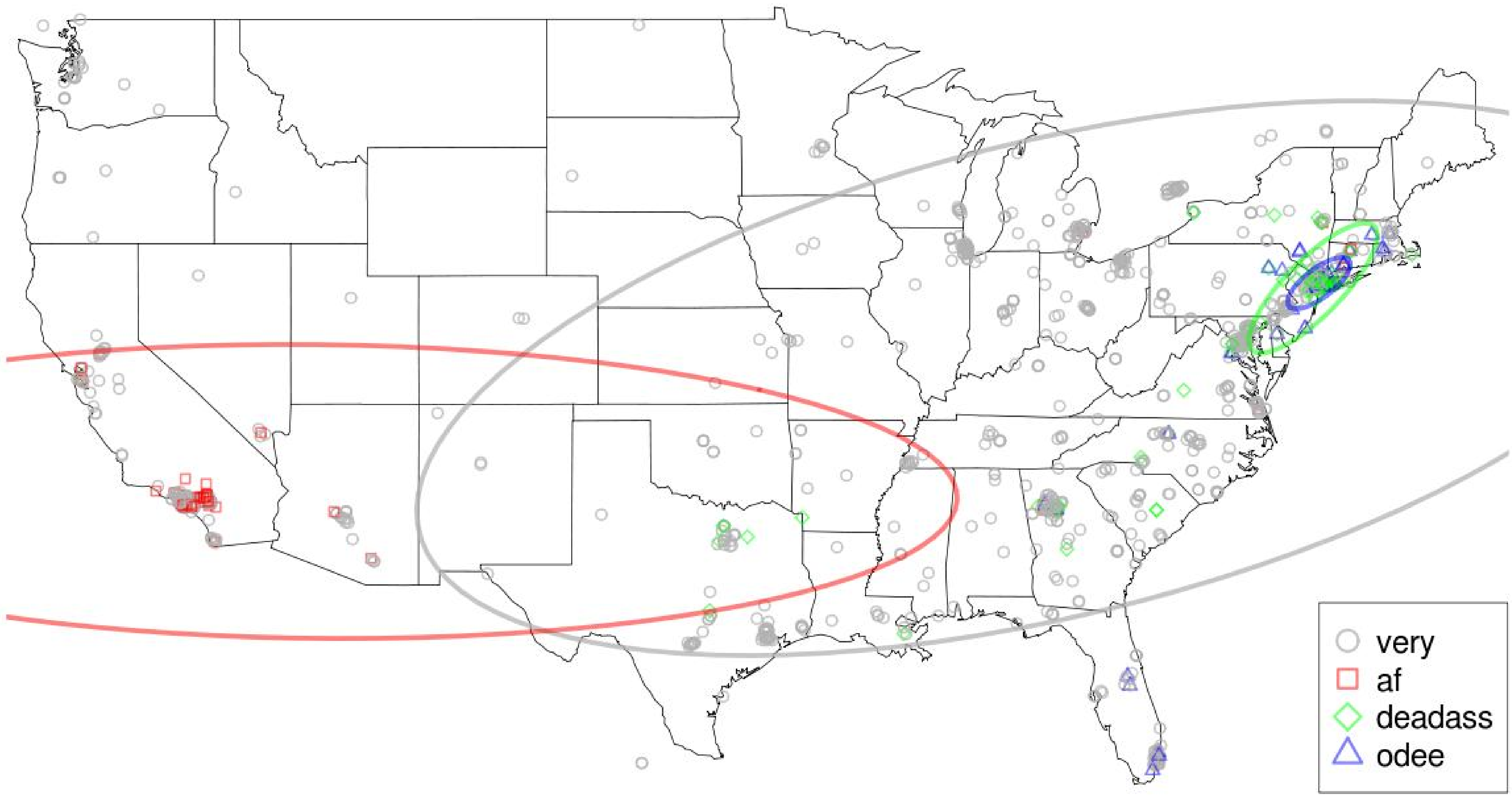
Social contingency of language



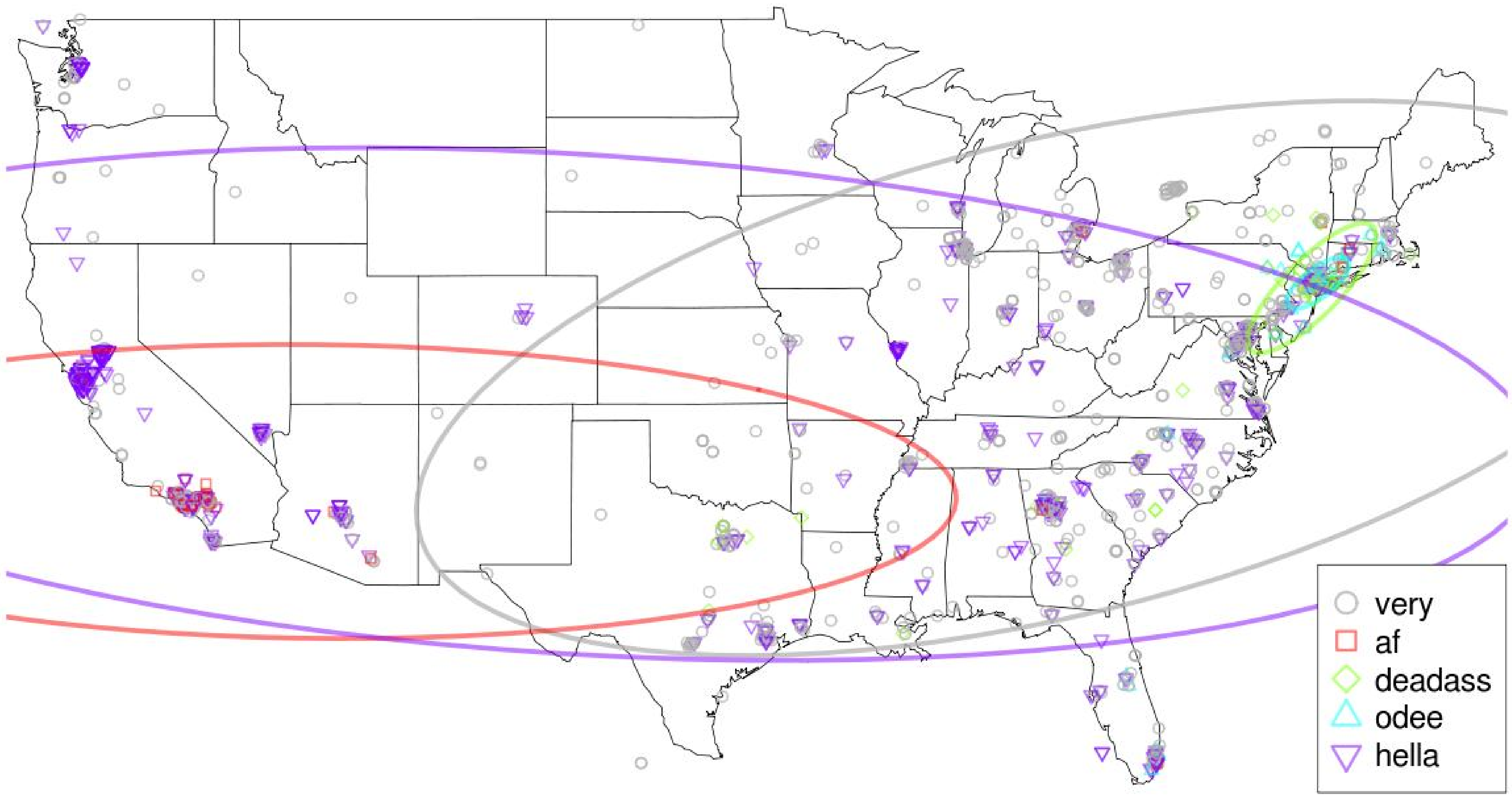
Social contingency of language



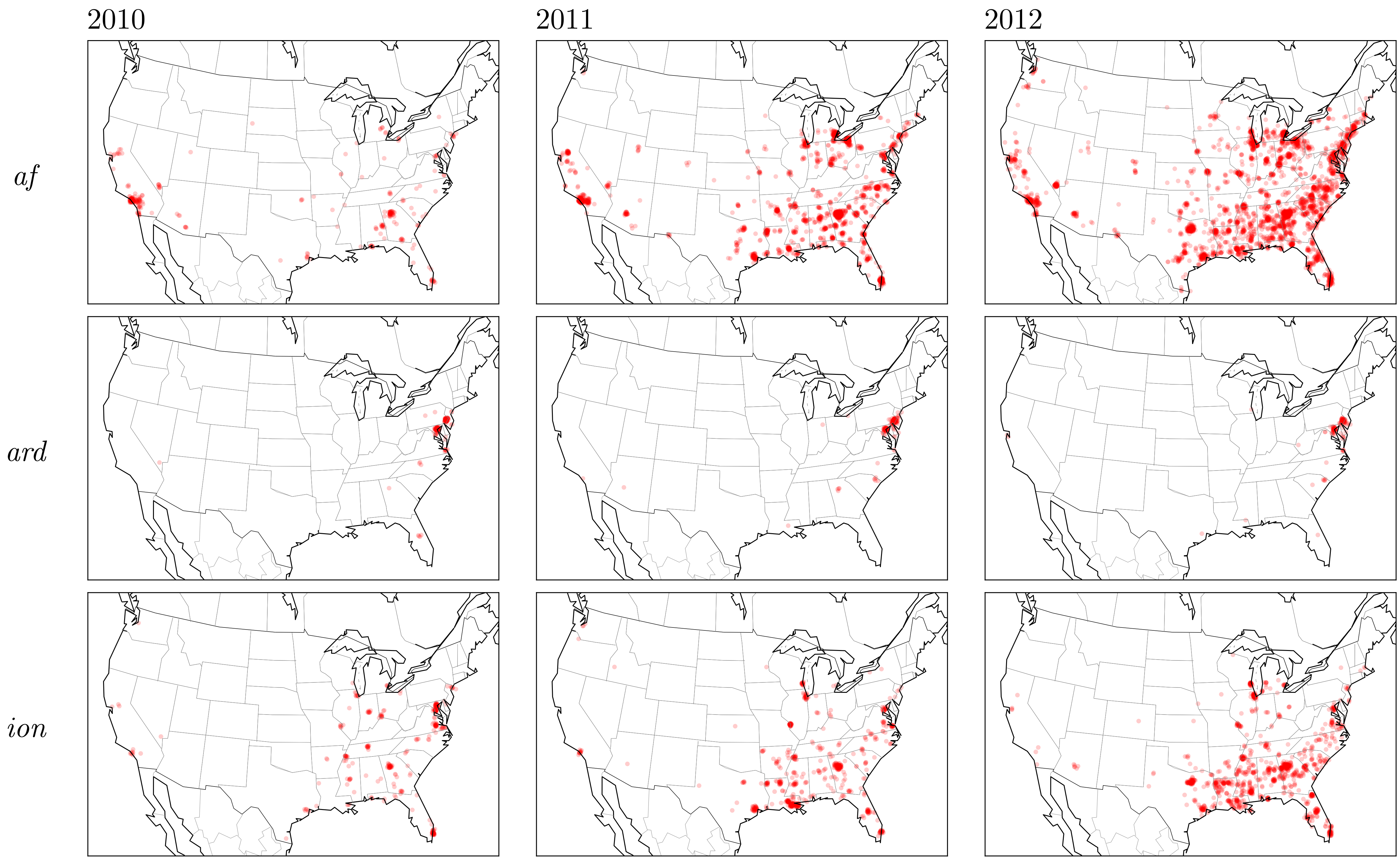
Social contingency of language



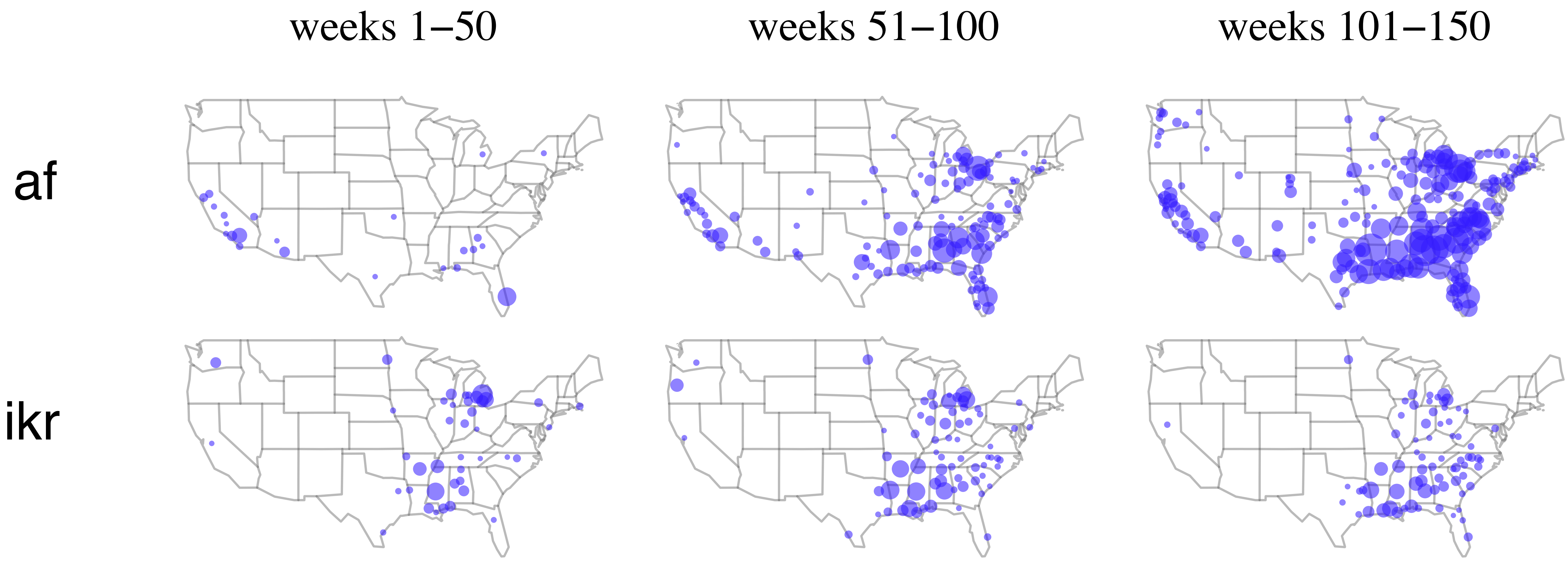
Social contingency of language



Social contingency of language



Social contingency of language



Alternate spellings

More remotely, *ard* is an alternative spelling for *alright*, as in:

(4) @name ard let me kno

(5) lol (*laugh out loud*) u'll be ard

Similarly, *brib* is an alternative spelling for *crib*, which in turn signifies *home*.

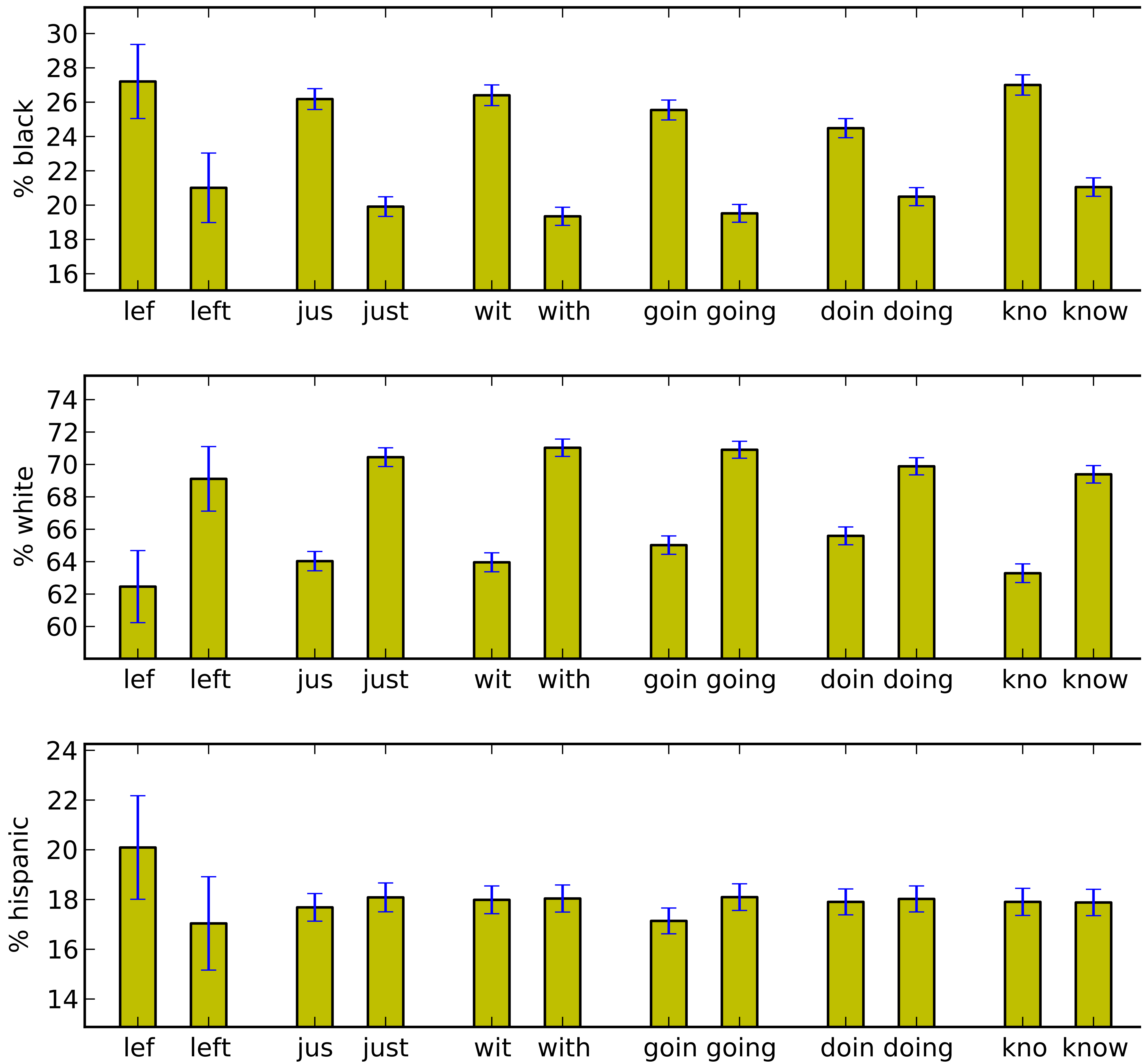
(6) bbq (*barbecue*) at my fams (*family's*) brib

(7) in da brib, just took a shower

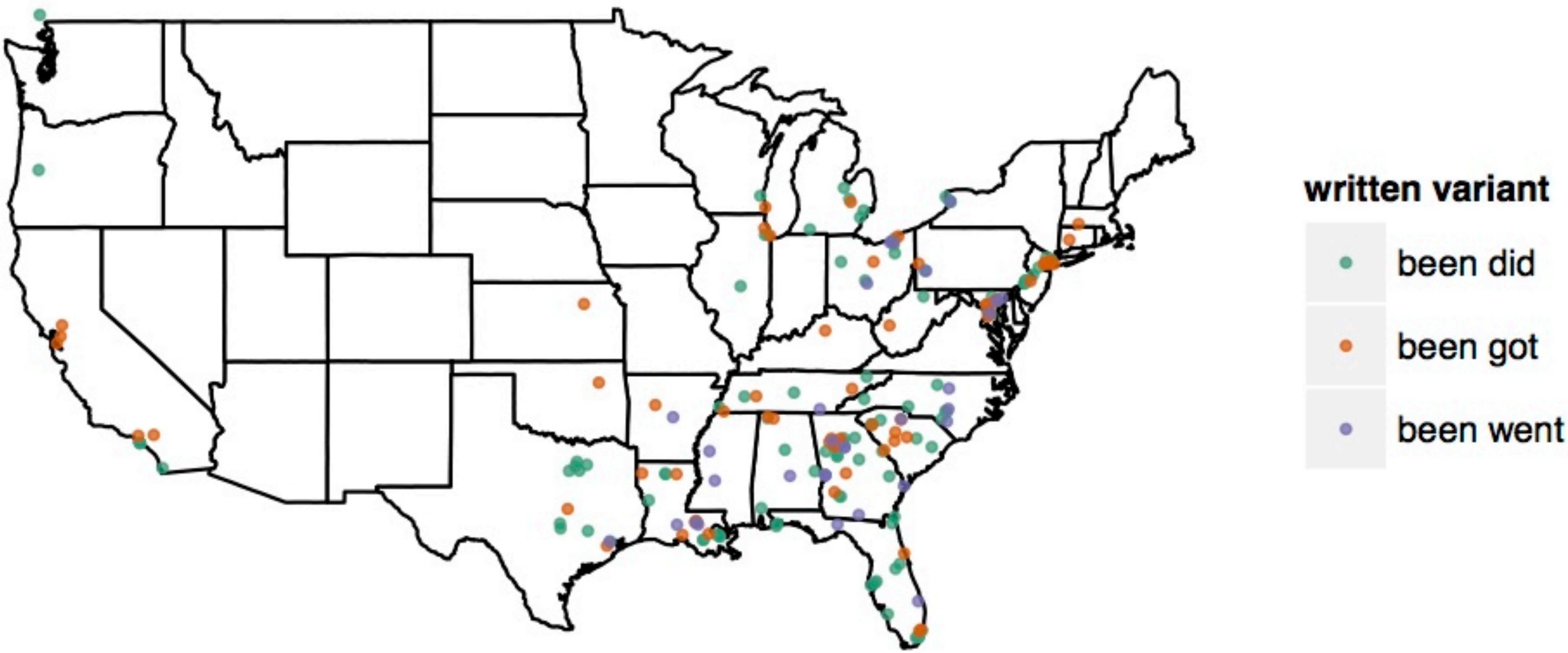
- Nationally, *brib* appears at a rate of once per 22,000 messages, which is roughly 5% as often as *crib*. But in the New York City area, *brib* appears at a rate of once per 3,000 messages.

Final consonant dropping

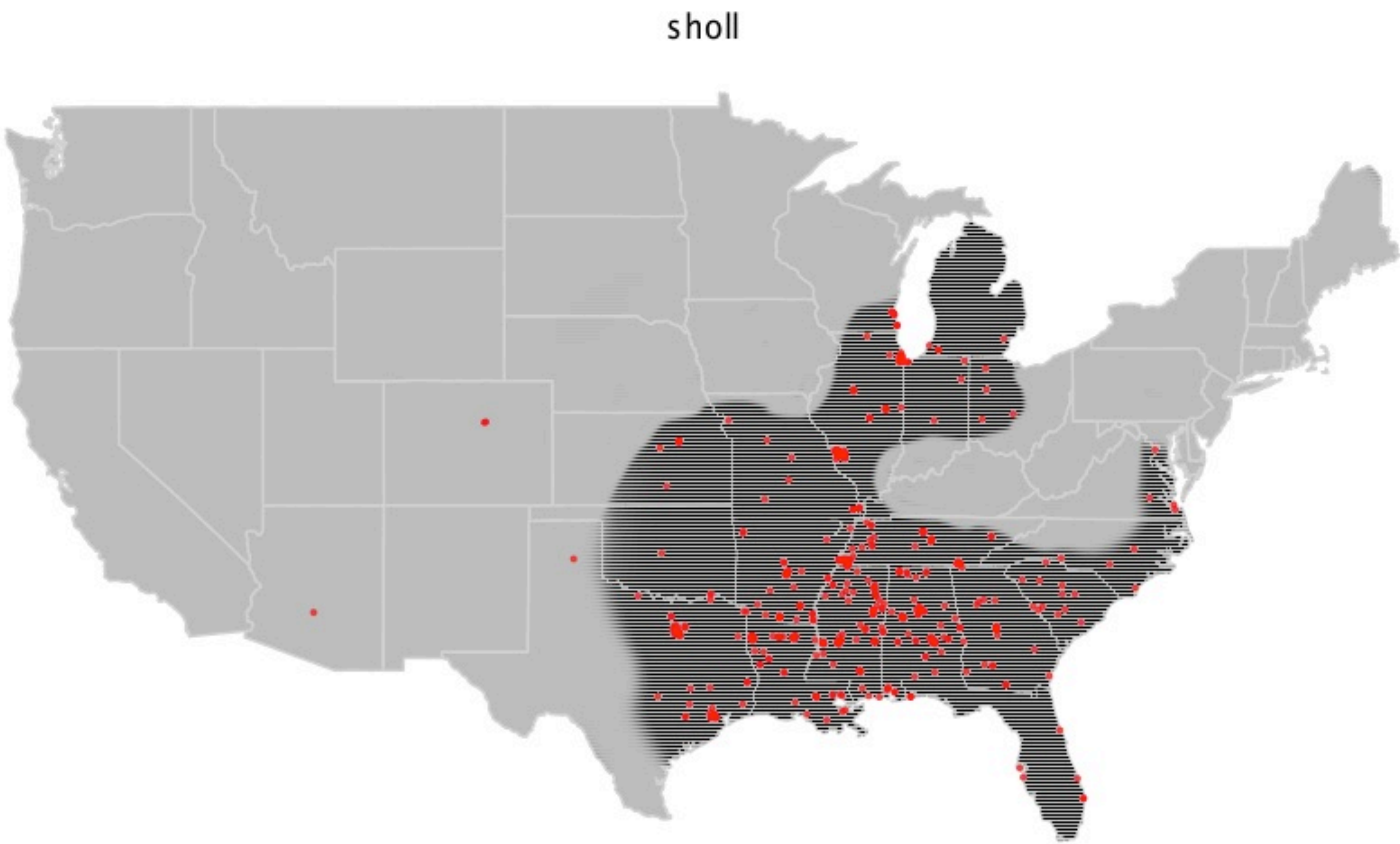
Avg. Census demographics of counties in which users of each term live



AAVE stressed 'been' on Twitter



From
http://www.languagejones.com/blog-1/2014/9/26/big-data-and-black-twitter



From
http://www.languagejones.com/blog-1/2014/9/26/big-data-and-black-twitter

From
http://www.languagejones.com/blog-1/2014/9/26/big-data-and-black-twitter

From Gouws et al. 2011 ... What drives alternate spellings? Typing UI?

Client	% In-Vocabulary
Facebook	88%
Twitter for iPhone	84%
Twitter for Blackberry	83%
Web	82%
UberTwitter	78%
Snaptu	73%
Overall	81%

Table 5: Percentage of in-vocabulary found in large English lexicon for different Twitter clients.

One interesting, but unexpected, finding is that the mobile phone (i.e., iPhone and Blackberry) clients have *fewer* out-of-vocabulary terms, on average, than the Web-based client. This suggests that either the users of the clients are less likely to misspell words or use slang terminology or that the clients may have better or more intuitive spell checking capabilities. A more thorough analysis is necessary to better understand the root cause of this phenomenon.

At the other end of the spectrum are the UberTwitter and Snaptu clients, which exhibit a substantially larger number of out-of-vocabulary terms. These clients are also typically used on mobile devices. As with our previous analysis, it is difficult to pinpoint the exact cause of such behavior, but we hypothesize that it is a function of user demographics and difficulties associated with inputting text on mobile devices.

NLP on social media's own terms

- Any NLP, starting with POS tagging, is going to require different models/resources than traditional written English

ikr	smh	he	asked	fir	yo	last
name	so	he	can	add	u	on
fb	lololol					

On a PTB-trained tagging model:
http://nlp.stanford.edu:8080/corenlp/process

- How to make a new POS tagger?
  - Annotate some data
  - Train a supervised sequence tagger
  - Have good features
  - Use semi-supervised learning to leverage unlabeled data

- Two examples: POS for Twitter
  - Ritter et al. 2011 (UW Twitter NLP)
  - Gimpel et al. 2011, Owoputi et al. 2013 (ARK TweetNLP)

Just a little annotated data

	#Msg.	#Tok.	Tagset	Dates
OCT27	1,827	26,594	App. A	Oct 27-28, 2010
DAILY547	547	7,707	App. A	Jan 2011–Jun 2012
NPSCHAT	10,578	44,997	PTB-like	Oct–Nov 2006
(w/o sys. msg.)	7,935	37,081		
RITTERTW	789	15,185	PTB-like	unknown

Table 1: Annotated datasets: number of messages, tokens, tagset, and date range. More information in §5, §6.3, and §6.2.

Features (MEMM tagger)

- Direct representations
  - Lexical identity
  - Shape features
  - Character n-gram prefix/suffix of word
- Regex detectors
  - Regex-based emoticon detectors
  - Regexes for hashtags, @-mentions
- Dictionary lookups
  - Traditional POS dictionary
  - Word clusters (next few slides)
- ...All of these at next/prev positions

- Does the algorithm matter?
  - First-order MEMM
  - Greedy decoding has same performance as Viterbi
    - Greedy decoding is 3 times faster, at least for us
  - CRF has slightly better performance (0.3% or so?)

Features (MEMM tagger)

```
private void initializeFeatureExtractors() throws IOException {
    allFeatureExtractors = new ArrayList<FeatureExtractorInterface>();

    allFeatureExtractors.add(new WordClusterPaths());
    allFeatureExtractors.add(new WordListFeatures.POSTagDict());
    allFeatureExtractors.add(new WordListFeatures.MetaphonePOSDict());

    allFeatureExtractors.add(new MiscFeatures.NgramSuffix(20));
    allFeatureExtractors.add(new MiscFeatures.NgramPrefix(20));
    allFeatureExtractors.add(new MiscFeatures.PrevWord());
    allFeatureExtractors.add(new MiscFeatures.NextWord());
    allFeatureExtractors.add(new MiscFeatures.WordformFeatures());

    allFeatureExtractors.add(new MiscFeatures.CapitalizationFeatures());
    allFeatureExtractors.add(new MiscFeatures.SimpleOrthFeatures());
    allFeatureExtractors.add(new MiscFeatures.PrevNext());

    allFeatureExtractors.add(new WordListFeatures.Listofnames("proper_names"));
    allFeatureExtractors.add(new WordListFeatures.Listofnames("celebs")); //2012-0
    allFeatureExtractors.add(new WordListFeatures.Listofnames("videogame")); //ju
    allFeatureExtractors.add(new WordListFeatures.Listofnames("mobyplaces"));    /
    allFeatureExtractors.add(new WordListFeatures.Listofnames("family"));
    allFeatureExtractors.add(new WordListFeatures.Listofnames("male"));
    allFeatureExtractors.add(new WordListFeatures.Listofnames("female"));

    allFeatureExtractors.add(new MiscFeatures.Positions());

    //allFeatureExtractors.add(new Prev2Words());
    //allFeatureExtractors.add(new Next2Words());
    //allFeatureExtractors.add(new MiscFeatures.URLFeatures());

}
```

Word clustering

- Unsupervised HMM to induce word classes. ("Brown clustering")
- Train on lots of unlabeled data
    - 56 M tweets, 847 M tokens
    - Compare to annotated data:
      3000 tweets, 30k tokens

Word clustering

Binary path	Top words (by frequency)
11101010100010	lmao lmfao lmaoo lmaooo hahahahaha lool ctfu rofl loool lmfaoo lmfaooo lmaoooo
11101010100011	haha hahaha hehe hahahaha hahah aha hehehe ahaha hah hahahah kk hahaa ah
11101010100100	yes yep yup nope yess yesss yessss ofcourse yeap likewise yepp yesh yw yuup yu
11101010100101	yeah yea nah naw yeahh nooo yeh noo noooo yeaa **ikr** nvm yeahhh nahh nooooo
1110101011100	**smh** jk #fail #random #fact smfh #smh #winning #realtalk smdh #dead #justsaying
011101011	**u** yu yuh yhu uu yuu yew y0u yuhh youh yhuu iget yoy yooh yuo ü yue juu Ʊ dya
1110010111001	w fo fa fr fro ov fer **fir** whit abou aft serie fore fah fuh w/her w/that fron isn agains
111101011000	facebook **fb** itunes myspace skype ebay tumblr bbm flickr aim msn netflix pandora
0011001	tryna gon finna bouta trynna boutta gne fina gonn tryina fenna qone trynaa qon
0011000	gonna gunna gona gna guna gnna ganna qonna gonnna gana qunna gonne goona
0110110111	soo sooo soooo sooooo soooooo sooooooo soooooooo sooooooooo soooooooooo
1110101001010	;) :p :-) xd ;-) ;d (; :3 ;p =p :-p =)) ;] xdd #gno xddd >:) ;-p >:d 8-) ;-d
1110101001011	:) (: =) :)) :] ☺ :') =] ^_^ :))) ^.^ [: ;)) 😊 ((: ^__^ (= ^-^ :))))
111010100111	:(:/ -_- -.- :-(:'(d: :| :s -__- =(=/ >.< -___- :-/ </3 :\ -_____- ;(/: :((>_< =
111010110001	<3 ♥ xoxo <33 xo <333 ❤ ♡ #love s2 <URL-twitition.com> #neversaynever

http://www.ark.cs.cmu.edu/TweetNLP/cluster_viewer.html

Word clusters as features

smh jk #fail #random #fact smfh #smh #winning #realtalk smdh #dead #justsaying

w fo fa fr fro ov fer **fir** whit abou aft serie fore fah fuh w/her w/that fron isn agains

yeah yea nah naw yeahh nooo yeh noo noooo yeaa **ikr** nvm yeahhh nahh nooooo

facebook **fb** itunes myspace skype ebay tumblr bbm flickr aim msn netflix pandora

ikr	smh	he	asked	fir	yo	last
!	G	O	V	P	D	A
name	so	he	can	add	u	on
N	P	O	V	V	O	P
fb	lololol					
^	!					

Clusters help a lot

Feature set	OCT27TEST	DAILY547	NPSCHATTEST	
All features	91.60	92.80	91.19	1
with clusters; without tagdicts, namelists	91.15	92.38	90.66	2
without clusters; with tagdicts, namelists	89.81	90.81	90.00	3
only clusters (and transitions)	89.50	90.54	89.55	4
without clusters, tagdicts, namelists	86.86	88.30	88.26	5
Gimpel et al. (2011) version 0.2	88.89	89.17		6
Inter-annotator agreement (Gimpel et al., 2011)	92.2			7
Model trained on all OCT27		93.2		8

[Ablation tests: remove a feature class, check performance]

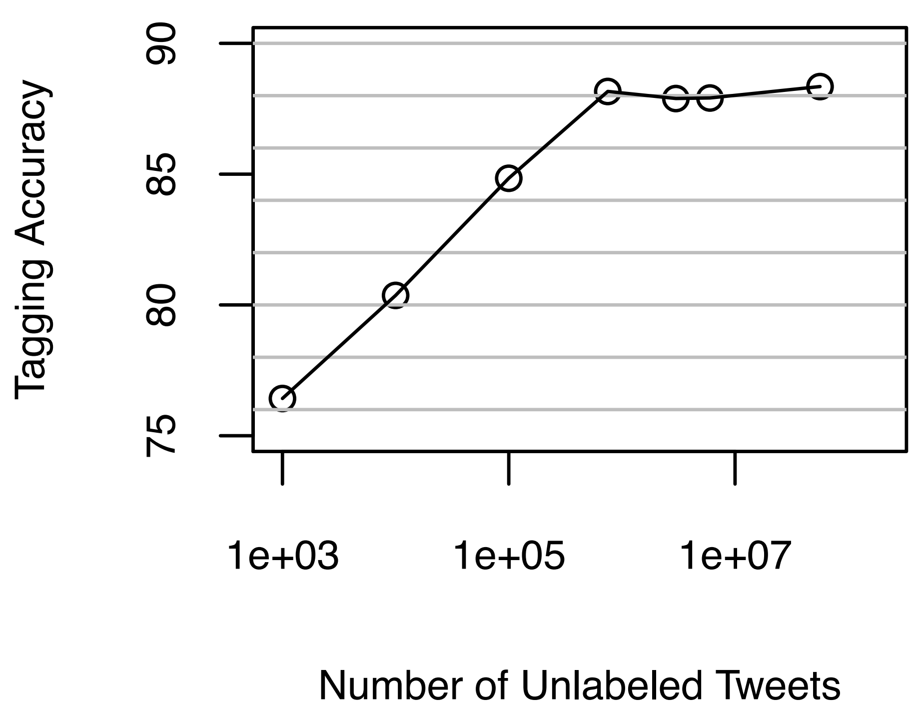


Figure 3: OCT27 development set accuracy using only clusters as features.

Clusters help for nonstandard terms

Model	**In dict.**	**Out of dict.**
Full	93.4	85.0
No clusters	92.0 (-1.4)	79.3 (-5.7)
Total tokens	*4,808*	*1,394*

Table 3: DAILY547 accuracies (%) for tokens in and out of a traditional dictionary, for models reported in rows 1 and 3 of Table 2.