Unlabeled data in NLP

CS 585, Fall 2017: Introduction to Natural Language Processing
http://people.cs.umass.edu/~brenocon/inlp2017

Brendan O’Connor
College of Information and Computer Sciences
University of Massachusetts Amherst

- Lots of unlabeled data, not much labeled data.
  How to use the unlabeled data?
- One trick: Learn **lexical** information (distributional/embeddings, first-order co-occurrence, etc.)
- More general ML settings
  - Unsupervised learning
  - Semi-supervised learning
- More general linguistic/knowledge structure settings
  - Relationships or events between entities
- Examples
  - EM algorithm: learning a generative model with <u>latent variables</u>
    - MNB/LDA document clusters, HMMs, translation...
  - Brown word clustering: a weird unsupervised HMM

Expectation-Maximization

- For latent-variable learning situations
  - **w**: known
  - **z**: unknown "nuisance" variable: need to infer
  - θ: want to learn
  - Learning goal: $\text{argmax}_\theta P(w \mid \theta) = \text{argmax}_\theta \Sigma_z P(w,z \mid \theta)$
- ... when parameter learning would be easy if only you had **z**.

- EM is a "meta"-algorithm
  - Initialize parameters.
  - Iterate until convergence (or stop early):
    - (E step): Infer $Q(z) := P(z \mid w, \theta)$
    - (M step): Learn new $\theta := \text{argmax}_\theta E_Q[\log P(w,z \mid \theta)]$
- "Bootstrapping" intuition
- It will converge to a local maximum solution to the original marginal likelihood learning goal

Doc categ/clustering

MNB

y: doc categ

$\vec{w}$: doc text

Model $P(\vec{w}, y) = P(y)\, P(\vec{w} | y)$

Sup. Learning: $\max_{\theta} P(w^{(tr)}, y^{(tr)})$

Unsup. Learning: $\max_{\theta} P(w^{(unlab)})$

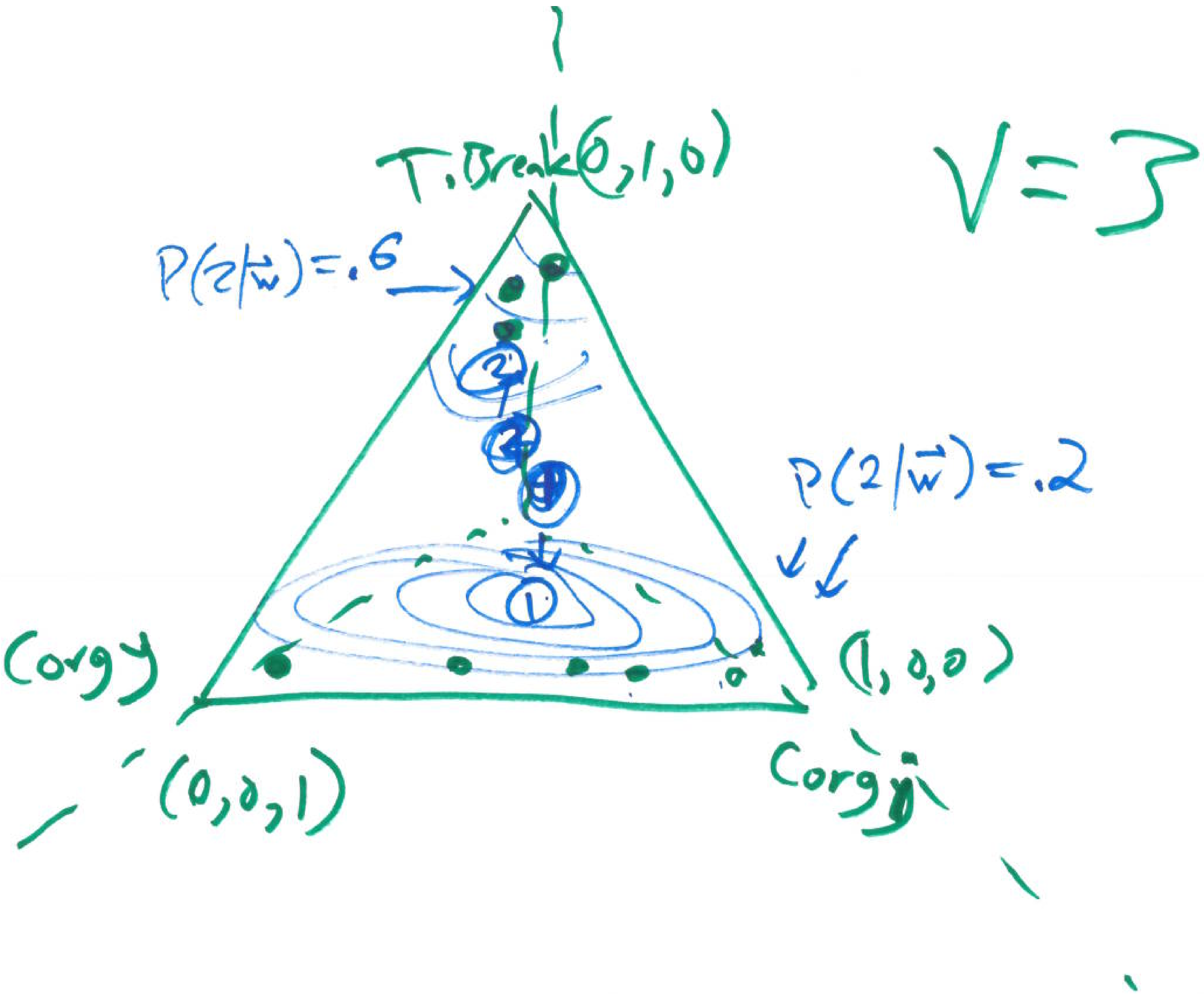
$$= \sum_y P(w^{(unlab)}, y)$$

$$P(\vec{w}^{(d)}, y^{(d)})$$

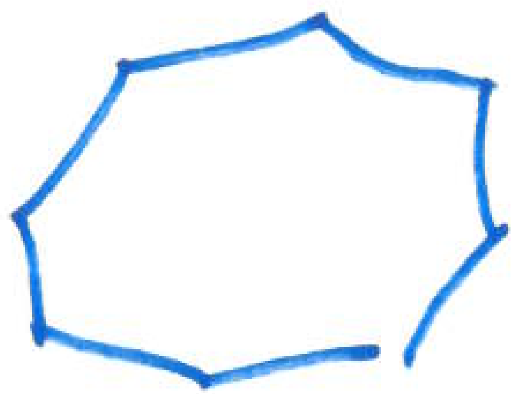
$$P(\vec{w}^{(d)}) = \sum_k P(y^{(d)} = k)\, P(\vec{w}^{(d)} | y^{(d)} = k)$$

EM: kinda like K-Means

but... for MNB (or more!)



V = 3



EM performance

- Guaranteed to find a locally-maximum solution. Guaranteed to converge.
- But can take a while
- Initialization-dependent

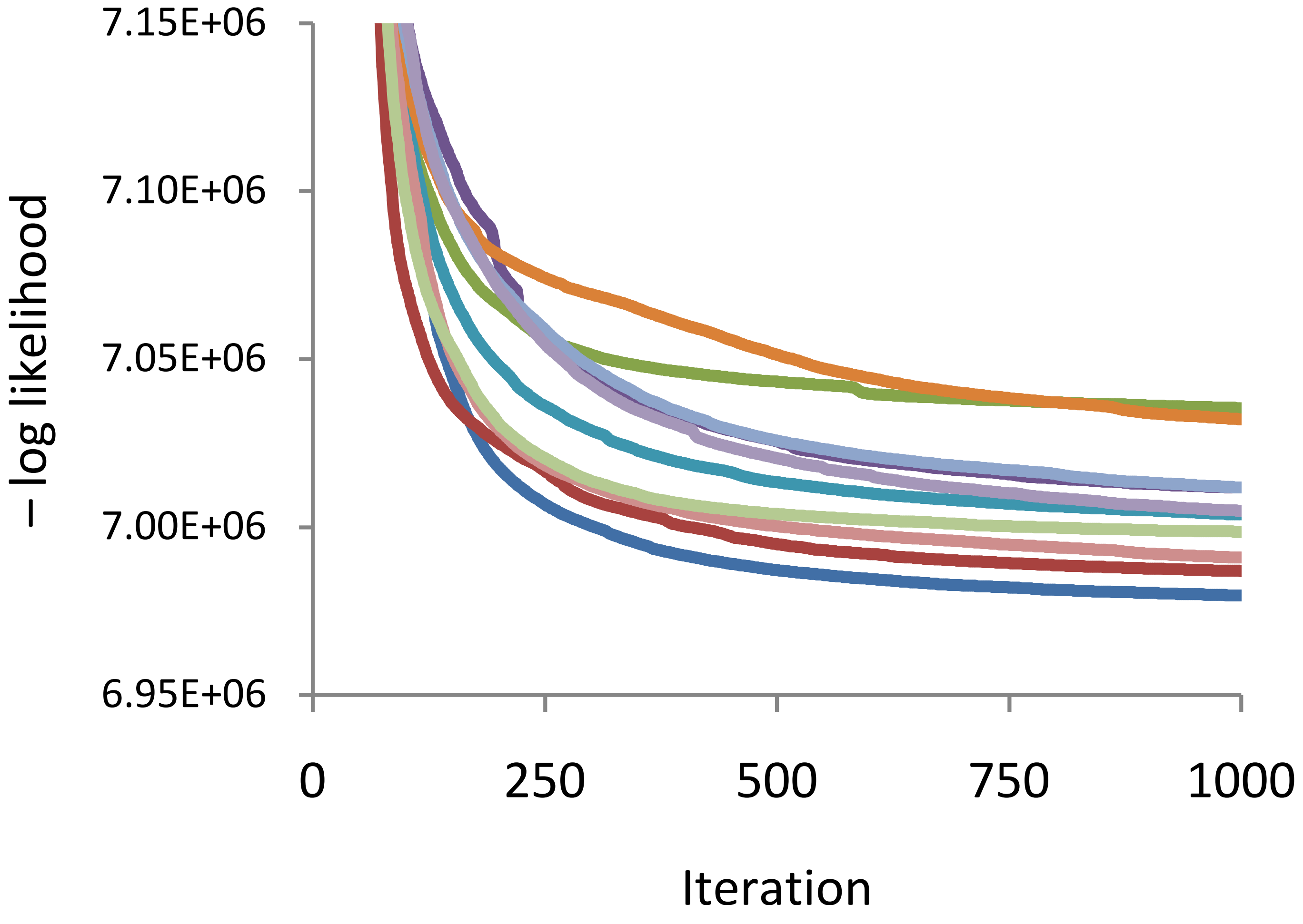


Figure 1: Variation in negative log likelihood with increasing iterations for 10 EM runs from different random starting points.

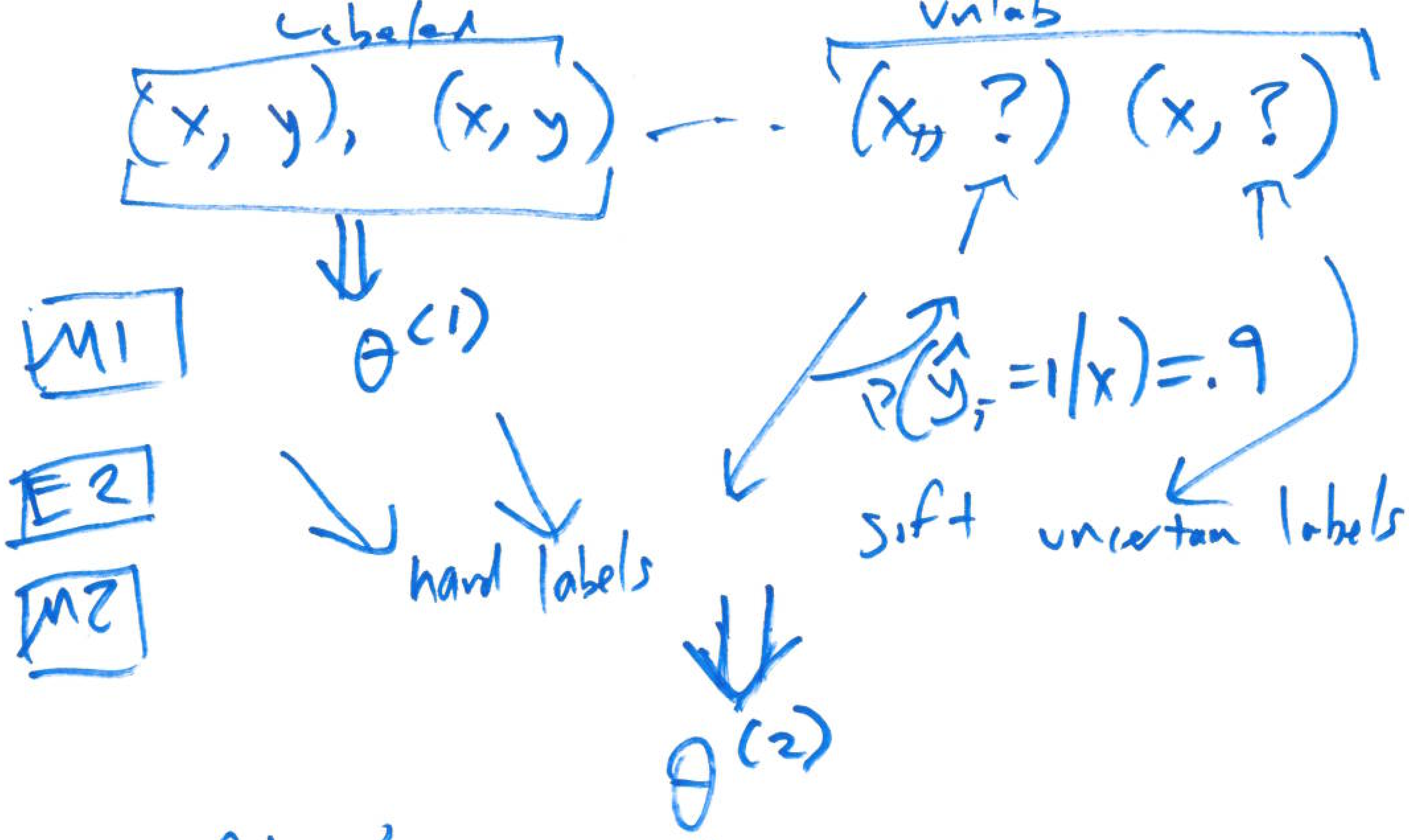
Johnson 2007, "Why doesn't EM find good HMM POS-taggers?"

Semi-supervised learning with EM

- “Semi-supervised”: *combine* unlabeled and labeled data

Semi-supervised learning with EM

- “Semi-supervised”: <u>*combine*</u> unlabeled and labeled data



Issues: overfit / get weird

Word embeddings/clusters as features

- Two-phase strategy
  - 1. Unsupervised learning of word representations (embeddings or clusters)
  - 2. Use word clusters as features for your small-data supervised model
- Word embeddings in a linear model
  - Turian et al. 2010: they work well in a CRF
  - Scaling issue: since they go alongside binary features
  - (IMO, they work even better in nonlinear models?)
- Or: Word clusters in a linear model

Assume that the embeddings are represented by a matrix E:

$$E \leftarrow \sigma \cdot E / stddev(E) \tag{1}$$

σ is a scaling constant that sets the new standard deviation after scaling the embeddings.

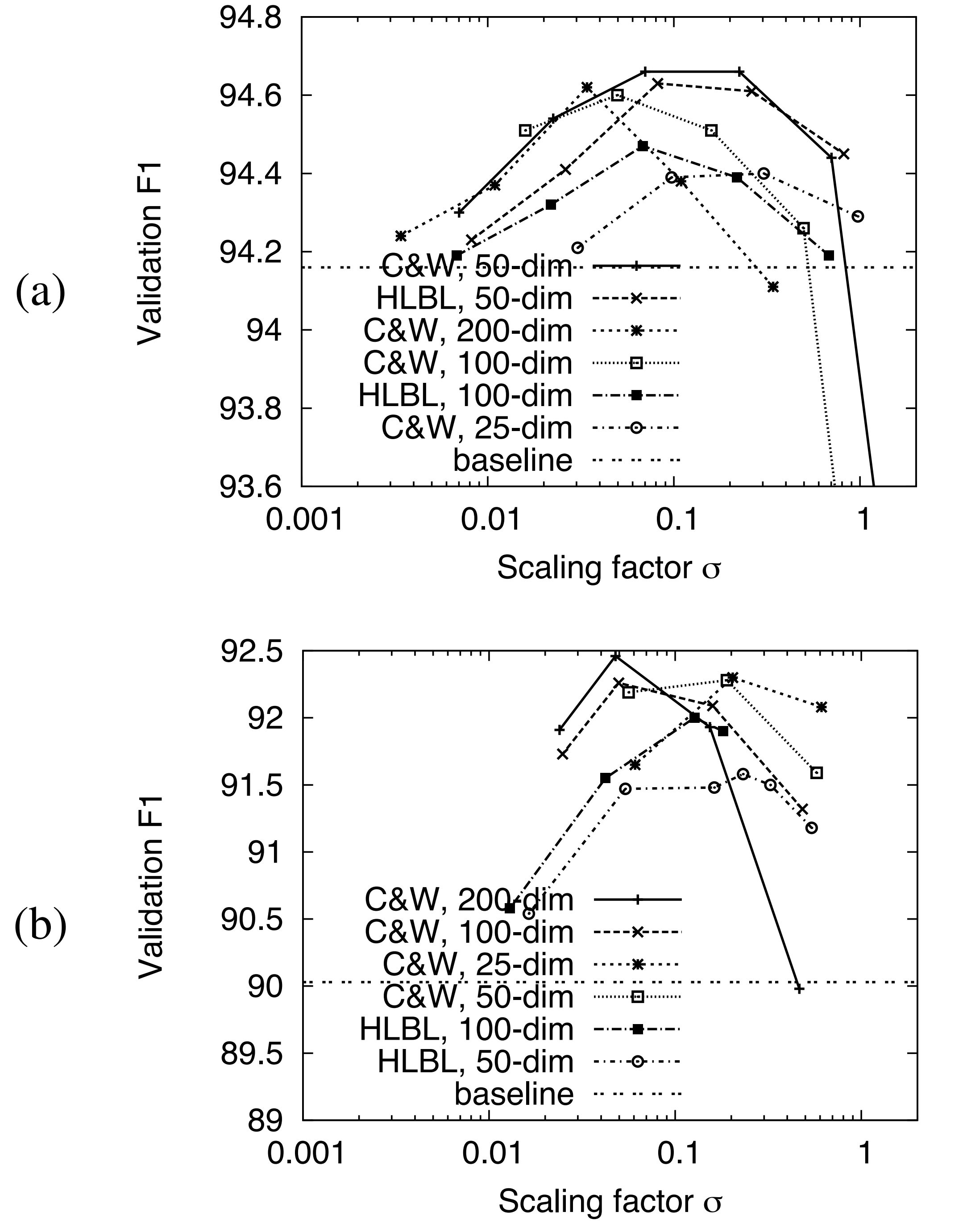


Figure 1: Effect as we vary the scaling factor σ (Equation 1) on the validation set F1. We experiment with Collobert and Weston (2008) and HLBL embeddings of various dimensionality. (a) Chunking results. (b) NER results.

Application: Social Media POS Tagging

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

- Any NLP system, starting with POS tagging, needs different models/resources than traditional written English
  - Annotate ~2300 tweets
  - Train word clusters on 56 million tweets, use as features

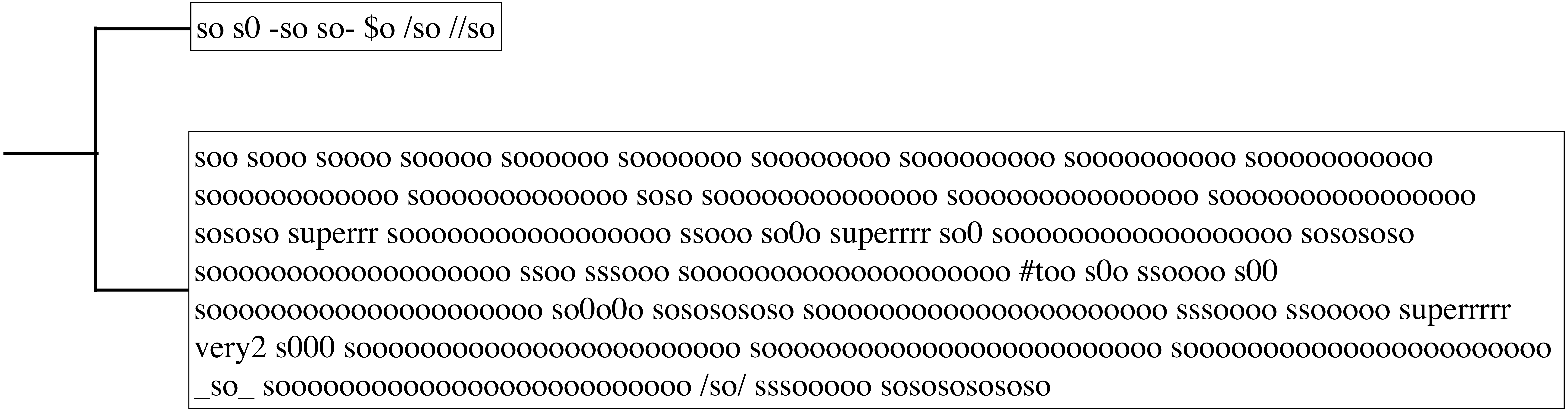
Hierarchical HMM-based word clustering ("Brown clustering")

- Only a little labeled data (2374 tweets)
- Lots of unlabeled data (56 million tweets): use for lexical generalization
- Distributional hypothesis:
  "you shall know a word by the company it keeps"
  - Unsupervised HMM with hierarchical clusters
    [*Percy Liang (2005)*'s version of Brown clustering]
  - 1000 clusters over 217k word types

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

What does it learn?

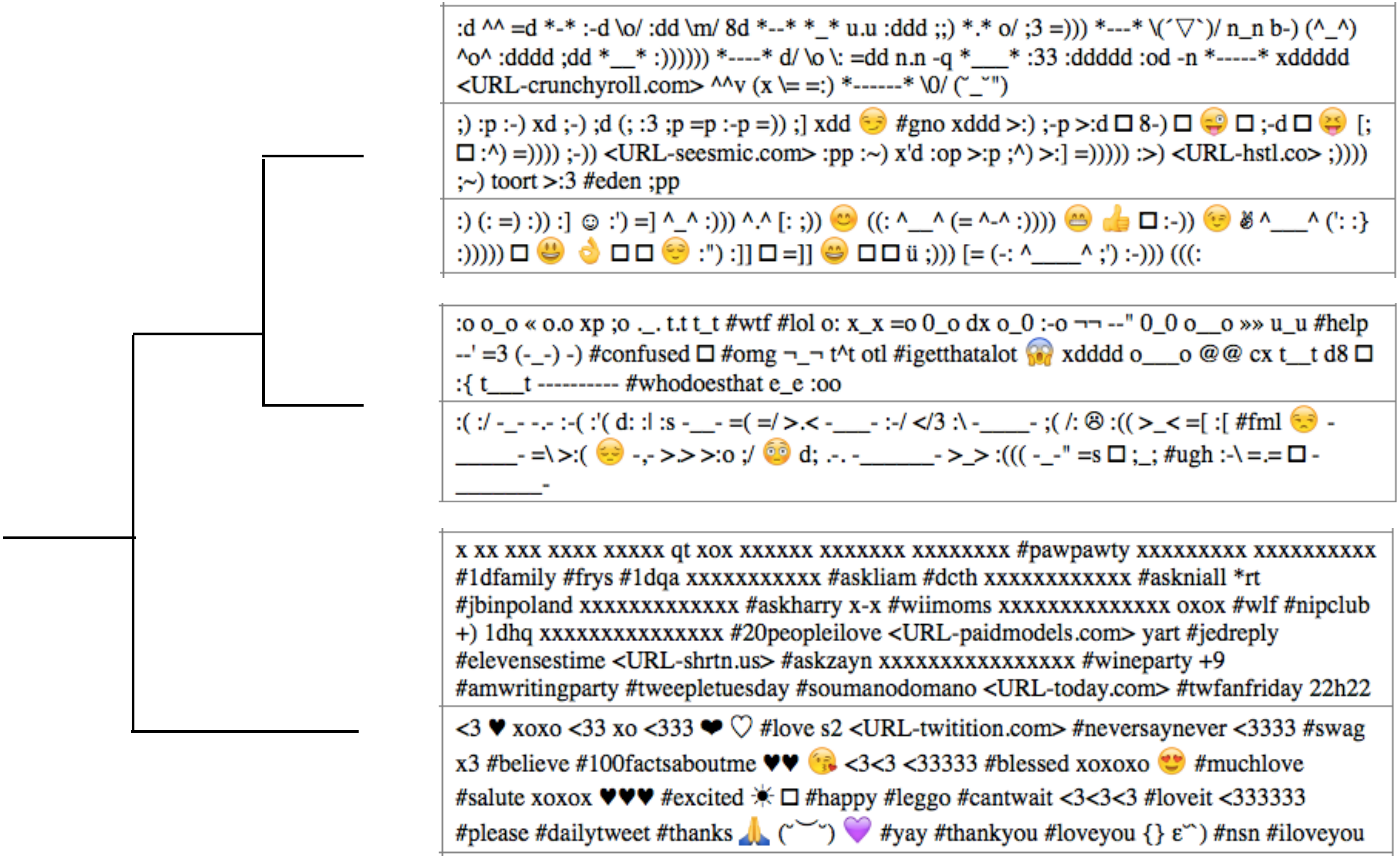
- Orthographic normalizations



so s0 -so so- $o /so //so

soo sooo soooo sooooo soooooo sooooooo soooooooo sooooooooo soooooooooo sooooooooooo soooooooooooo sooooooooooooo soso soooooooooooooo sooooooooooooooo soooooooooooooooo sososo superrr sooooooooooooooooo ssooo so0o superrrr so0 soooooooooooooooooo sosososo sooooooooooooooooooo ssoo sssooo soooooooooooooooooooo #too s0o ssoooo s00 sooooooooooooooooooooo so0o0o sososososo soooooooooooooooooooooo sssoooo ssooooo superrrrr very2 s000 sooooooooooooooooooooooo soooooooooooooooooooooooo sooooooooooooooooooooooooo _so_ soooooooooooooooooooooooooo /so/ sssooooo sosososososo

- Emoticons etc.
(Clusters/tagger useful for sentiment analysis: NRC-Canada SemEval 2013, 2014)



:d ^^ =d *-* :-d \o/ :dd \m/ 8d *--* *_* u.u :ddd ;;) *.* o/ ;3 =))) *---* \(´▽`)/ n_n b-) (^_^) ^o^ :dddd ;dd *__* :)))))) *----* d/ \o \: =dd n.n -q *___* :33 :ddddd :od -n *-----* xddddd <URL-crunchyroll.com> ^^v (x \= =:) *------* \0/ (˘_˘")

;) :p :-) xd ;-) ;d (; :3 ;p =p :-p =)) ;] xdd 😏 #gno xddd >:) ;-p >:d □ 8-) □ 😜 □ ;-d □ 😝 [; □ :^) =)))) ;-)) <URL-seesmic.com> :pp :~) x'd :op >:p ;^) >:] =))))) :>) <URL-hstl.co> ;)))) ;~) toort >:3 #eden ;pp

:) (: =) :)) :] ☺ :') =] ^_^ :))) ^.^ [: ;)) 😊 ((: ^__^ (= ^-^ :)))) 😁 👍 □ :-)) 😉 ✌ ^___^ (': :} :))))) □ 😃 👌 □ □ 😌 :") :]] □ =]] 😄 □ □ ü ;))) [= (-: ^____^ ;') :-))) (((:

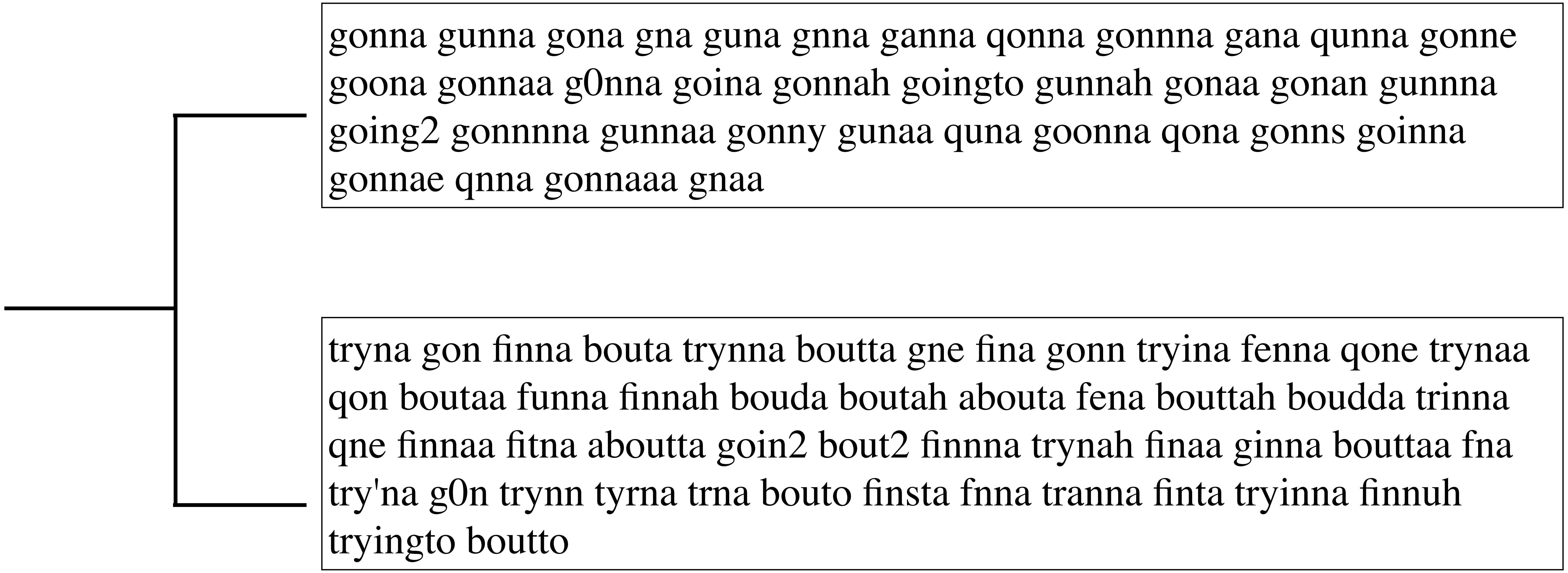
:o o_o « o.o xp ;o ._. t.t t_t #wtf #lol o: x_x =o 0_o dx o_0 :-o ¬¬ --" 0_0 o__o »» u_u #help --' =3 (-_-) -) #confused □ #omg ¬_¬ t^t otl #igetthatalot 😱 xdddd o___o @@ cx t__t d8 □ :{ t___t ---------- #whodoesthat e_e :oo

:(:/ -_- -.- :-(:'(d: :| :s -__- =(=/ >.< -___- :-/ </3 :\ -____- ;(/: ☹ :((>_< =[:[#fml 😒 -_____- =\ >:(😔 -,- >.> >:o ;/ 😳 d; .-. -______- >_> :(((-_-" =s □ ;_; #ugh :-\ =.= □ -_______-

x xx xxx xxxx xxxxx qt xox xxxxxx xxxxxxx xxxxxxxx #pawpawty xxxxxxxxx xxxxxxxxxx #1dfamily #frys #1dqa xxxxxxxxxxx #askliam #dcth xxxxxxxxxxxx #askniall *rt #jbinpoland xxxxxxxxxxxxx #askharry x-x #wiimoms xxxxxxxxxxxxxx oxox #wlf #nipclub +) 1dhq xxxxxxxxxxxxxxx #20peopleilove <URL-paidmodels.com> yart #jedreply #elevensestime <URL-shrtn.us> #askzayn xxxxxxxxxxxxxxxx #wineparty +9 #amwritingparty #tweepletuesday #soumanodomano <URL-today.com> #twfanfriday 22h22

<3 ♥ xoxo <33 xo <333 ❤ ♡ #love s2 <URL-twitition.com> #neversaynever <3333 #swag x3 #believe #100factsaboutme ♥♥ 😘 <3<3 <33333 #blessed xoxoxo 😍 #muchlove #salute xoxox ♥♥♥ #excited ☀ □ #happy #leggo #cantwait <3<3<3 #loveit <333333 #please #dailytweet #thanks 🙏 (˘︶˘) 💜 #yay #thankyou #loveyou {} ε˘`) #nsn #iloveyou

(Immediate?) future auxiliaries

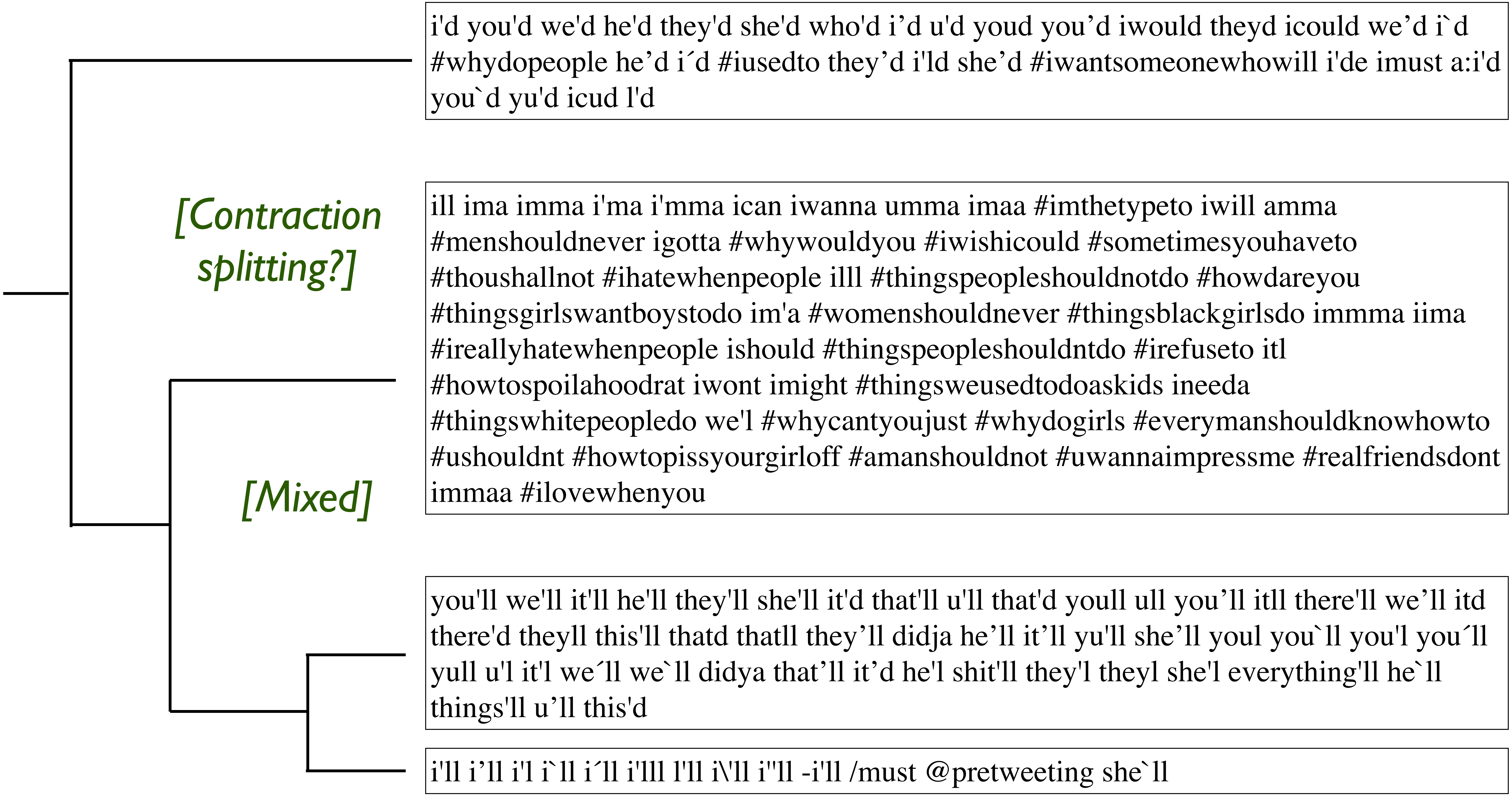


gonna gunna gona gna guna gnna ganna qonna gonnna gana qunna gonne goona gonnaa g0nna goina gonnah goingto gunnah gonaa gonan gunnna going2 gonnnna gunnaa gonny gunaa quna goonna qona gonns goinna gonnae qnna gonnaaa gnaa

tryna gon finna bouta trynna boutta gne fina gonn tryina fenna qone trynaa qon boutaa funna finnah bouda boutah abouta fena bouttah boudda trinna qne finnaa fitna aboutta goin2 bout2 finnna trynah finaa ginna bouttaa fna try'na g0n trynn tyrna trna bouto finsta fnna tranna finta tryinna finnuh tryingto boutto

- finna ~ “fixing to”
- tryna ~ “trying to”
- bouta ~ “about to”

Subject-AuxVerb constructs



i'd you'd we'd he'd they'd she'd who'd i’d u'd youd you’d iwould theyd icould we’d i`d #whydopeople he’d i´d #iusedto they’d i'ld she’d #iwantsomeonewhowill i'de imust a:i'd you`d yu'd icud l'd

[Contraction splitting?]

ill ima imma i'ma i'mma ican iwanna umma imaa #imthetypeto iwill amma #menshouldnever igotta #whywouldyou #iwishicould #sometimesyouhaveto #thoushallnot #ihatewhenpeople illl #thingspeopleshouldnotdo #howdareyou #thingsgirlswantboystodo im'a #womenshouldnever #thingsblackgirlsdo immma iima #ireallyhatewhenpeople ishould #thingspeopleshouldntdo #irefuseto itl #howtospoilahoodrat iwont imight #thingsweusedtodoaskids ineeda #thingswhitepeopledo we'l #whycantyoujust #whydogirls #everymanshouldknowhowto #ushouldnt #howtopissyourgirloff #amanshouldnot #uwannaimpressme #realfriendsdont immaa #ilovewhenyou

[Mixed]

you'll we'll it'll he'll they'll she'll it'd that'll u'll that'd youll ull you’ll itll there'll we’ll itd there'd theyll this'll thatd thatll they’ll didja he’ll it’ll yu'll she’ll youl you`ll you'l you´ll yull u'l it'l we´ll we`ll didya that’ll it’d he'l shit'll they'l theyl she'l everything'll he`ll things'll u’ll this'd

i'll i’ll i'l i`ll i´ll i'lll l'll i\'ll i''ll -i'll /must @pretweeting she`ll

Application: Social Media POS Tagging

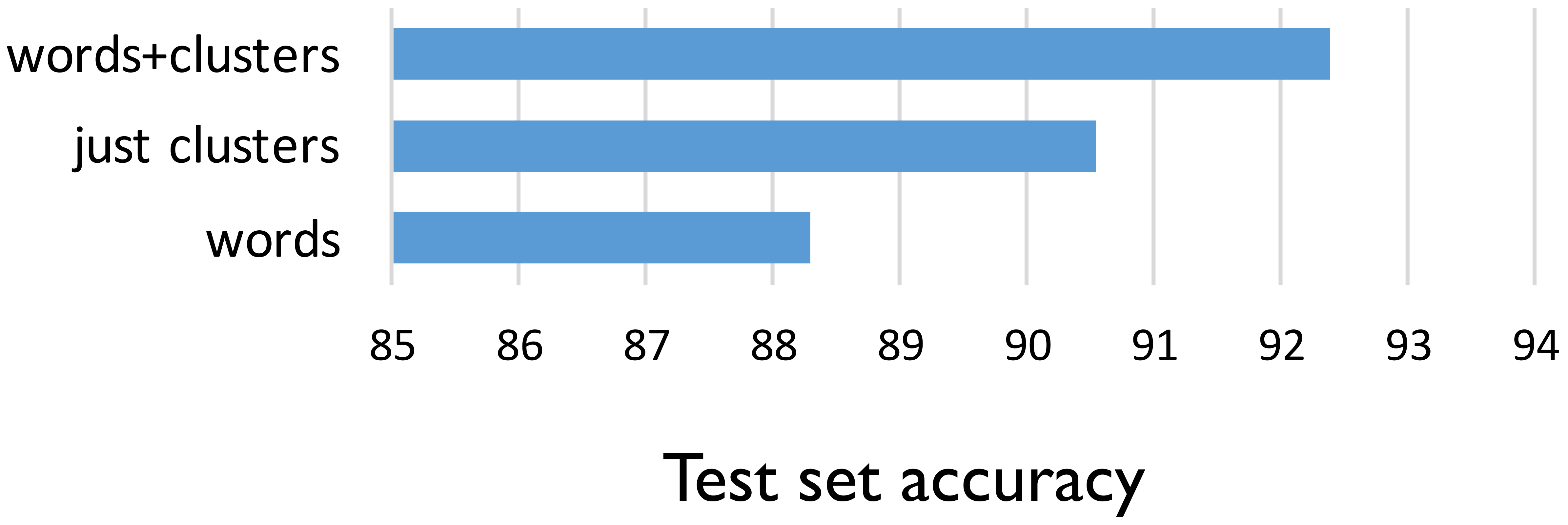
ikr	smh	he	asked	fir	yo	last
!	G	O	V	P	D	A
name	so	he	can	add	u	on
N	P	O	V	V	O	P
fb	lololol					
^	!					

w fo fa fr fro ov fer **fir** whit abou aft serie fore fah fuh w/her w/that fron isn agains	“non-standard prepositions”
yeah yea nah naw yeahh nooo yeh noo noooo yeaa **ikr** nvm yeahhh nahh nooooo	“interjections”
facebook **fb** itunes myspace skype ebay tumblr bbm flickr aim msn netflix pandora	“online service names”
smh jk #fail #random #fact smfh #smh #winning #realtalk smdh #dead #justsaying	“hashtag-y interjections”??

Highest-weighted POS–treenode features hierarchical structure gives multiresolutional generalization

Cluster prefix	Tag	Types	Most common word in each cluster with prefix
11101010*	!	8160	lol lmao haha yes yea oh omg aww ah btw wow thanks sorry congrats welcome yay ha hey goodnight hi dear please huh wtf exactly idk bless whatever well ok
11000*	L	428	i'm im you're we're he's there's its it's
1110101100*	E	2798	x <3 :d :p :) :o :/
111110*	A	6510	young sexy hot slow dark low interesting easy important safe perfect special different random short quick bad crazy serious stupid weird lucky sad
1101*	D	378	the da my your ur our their his
01*	V	29267	do did kno know care mean hurts hurt say realize believe worry understand forget agree remember love miss hate think thought knew hope wish guess bet have
11101*	O	899	you yall u it mine everything nothing something anyone someone everyone nobody
100110*	&	103	or n & and

Clusters help POS tagging



“words”: all handcrafted features

- A little annotation + lots of unlabeled data
- Unsupervised word representation learning (clusters, embeddings) is a crucial technique in NLP