Supervised sentiment analysis: General practical tips

Christopher Potts

Stanford Linguistics

CS224u: Natural language understanding

Selected sentiment datasets

There are too many to try to list, so I picked some with noteworthy properties, limiting to the core task of sentiment analysis:

- IMDb movie reviews (50K) (Maas et al. 2011): http://ai.stanford.edu/~amaas/data/sentiment/index.html
- Datasets from Lillian Lee's group: http://www.cs.cornell.edu/home/llee/data/
- Datasets from Bing Liu's group: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
- Amazon Customer Review data: https://s3.amazonaws.com/amazon-reviews-pds/readme.html
- Amazon Product Data (McAuley et al. 2015; He and McAuley 2016): http://jmcauley.ucsd.edu/data/amazon/
- Sentiment and social networks together (West et al. 2014) http://infolab.stanford.edu/~west1/TACL2014/
- Stanford Sentiment Treebank (SST; Socher et al. 2013) https://nlp.stanford.edu/sentiment/
- DynaSent (Potts et al. 2020) https://github.com/cgpotts/dynasent/

Lexica

- Bing Liu's Opinion Lexicon: `nltk.corpus.opinion_lexicon`
- SentiWordNet: `nltk.corpus.sentiwordnet`
- MPQA subjectivity lexicon: http://mpqa.cs.pitt.edu
- Harvard General Inquirer
  - Download: http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm
  - Documentation: http://www.wjh.harvard.edu/~inquirer/homecat.htm
- Linguistic Inquiry and Word Counts (LIWC): https://liwc.wpengine.com
- Hamilton et al. (2016): SocialSent https://nlp.stanford.edu/projects/socialsent/
- Brysbaert et al. (2014): Norms of valence, arousal, and dominance for 13,915 English lemmas

Tokenizing

Raw text

@NLUers: can't wait for the Jun 9 #projects!
YAAAAAAY!!! >:-D http://stanford.edu/class/cs224u/.

Isolate mark-up, and replace HTML entities.

@NLUers: can’t wait for the Jun 9 #projects! YAAAAAAY!!!
>:-D http://stanford.edu/class/cs224u/.

Whitespace tokenizer

Raw text

@NLUers: can't wait for the Jun 9 #projects! YAAAAAAY!!! >:-D http://stanford.edu/class/cs224u/.

@NLUers:
can’t
wait
for
the
Jun
9
#projects
YAAAAAAY!!!
>:-D
http://stanford.edu/class/cs224u/.

Treebank tokenizer

Raw text

@NLUers: can't wait for the Jun 9 #projects! YAAAAAAY!!! >:-D http://stanford.edu/class/cs224u/.

@	!
NLUers	YAAAAAAY
:	!
ca	!
n't	!
wait	>
for	:
the	-D
Jun	http
9	:
#	//stanford.edu/class/cs224u/
projects	.

Sentiment-aware tokenizer

- Isolates emoticons
- Respects Twitter and other domain-specific markup
- Uses the underlying mark-up (e.g., <strong> tags)
- Captures those #$%ing masked curses!
- Preserves capitalization where it seems meaningful
- Regularizes lengthening (e.g., *YAAAAAAY*⇒*YAAAY*)
- Captures significant multiword expressions (e.g., *out of this world*)

Sentiment-aware tokenizer

Raw text

@NLUers: can't wait for the Jun 9 #projects! YAAAAAAY!!! >:-D http://stanford.edu/class/cs224u/.

@nluers	!
:	YAAAY
can’t	!
wait	!
for	!
the	>:-D
Jun_9	http://stanford.edu/class/cs224u/
#projects	.

Sentiment-aware tokenizer

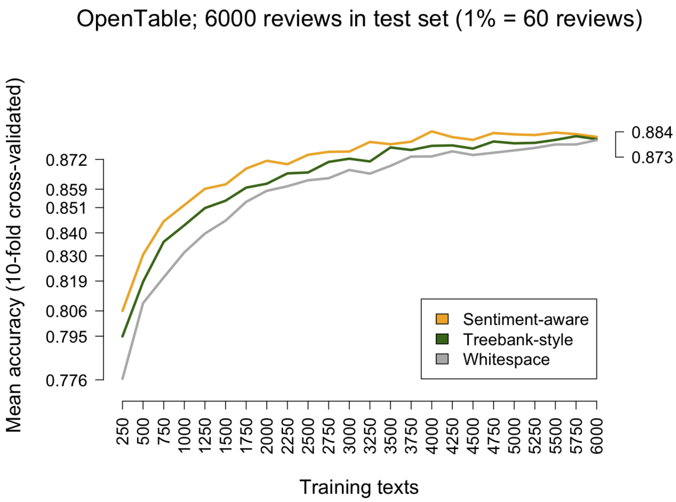
Raw text

@NLUers: can't wait for the Jun 9 #projects!
YAAAAAAY!!! >:-D http://stanford.edu/class/cs224u/.

@nluers	!
:	YAAAY
can’t	!
wait	!
for	!
the	>:-D
Jun_9	http://stanford.edu/class/cs224u/
#projects	.

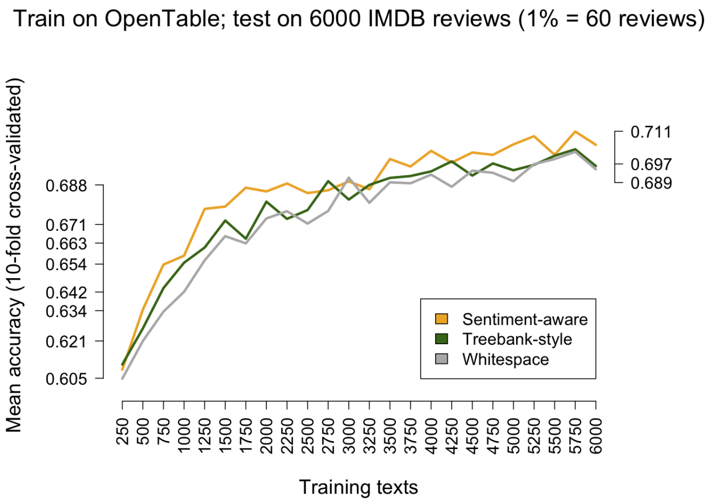
A good start: `nltk.tokenize.casual.TweetTokenizer`

The impact of sentiment-aware tokenizing



Softmax classifier.

The impact of sentiment-aware tokenizing



Softmax classifier.

The dangers of stemming

- Stemming collapses distinct word forms.
- Three common stemming algorithms:
  - the Porter stemmer
  - the Lancaster stemmer
  - the WordNet stemmer
- Porter and Lancaster destroy too many sentiment distinctions.
- The WordNet stemmer does not have this problem nearly so severely, but it generally doesn’t do enough collapsing to be worth the resources necessary to run it.

The Porter stemmer

Heuristically identifies word suffixes (endings) and strips them off, with some regularization of the endings.

Positiv	Negativ	Porter stemmed
defense	defensive	defens
extravagance	extravagant	extravag
affection	affectation	affect
competence	compete	compet
impetus	impetuous	impetu
objective	objection	object
temperance	temper	temper
tolerant	tolerable	toler

Table: Sample of instances in which the Porter stemmer destroys a Harvard Inquirer Positiv/Negativ distinction.

The Lancaster stemmer

Uses the same strategy as the Porter stemmer.

Positiv	Negativ	Lancaster stemmed
call	callous	cal
compliment	complicate	comply
dependability	dependent	depend
famous	famished	fam
fill	filth	fil
flourish	floor	flo
notoriety	notorious	not
passionate	passe	pass
savings	savage	sav
truth	truant	tru

Table: Sample of instances in which the Lancaster stemmer destroys a Harvard Inquirer Positiv/Negativ distinction.

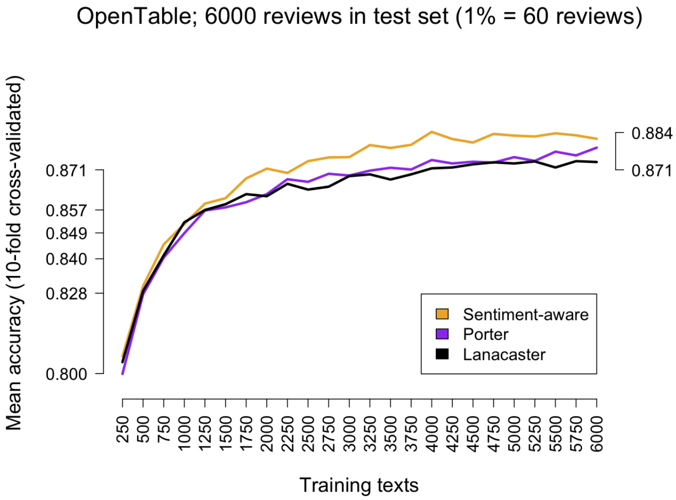
The WordNet stemmer

The WordNet stemmer (NLTK) is high-precision. It requires word–POS pairs. Its only general issue for sentiment is that it removes comparative morphology.

Positiv	WordNet stemmed
(exclaims, v)	exclaim
(exclaimed, v)	exclaim
(exclaiming, v)	exclaim
(exclamation, n)	exclamation
(proved, v)	prove
(proven, v)	prove
(proven, a)	proven
(happy, a)	happy
(happier, a)	happy
(happiest, a)	happy

Table: Representative examples of what WordNet stemming does and doesn't do.

The impact of stemming



Softmax classifier.

Other preprocessing techniques

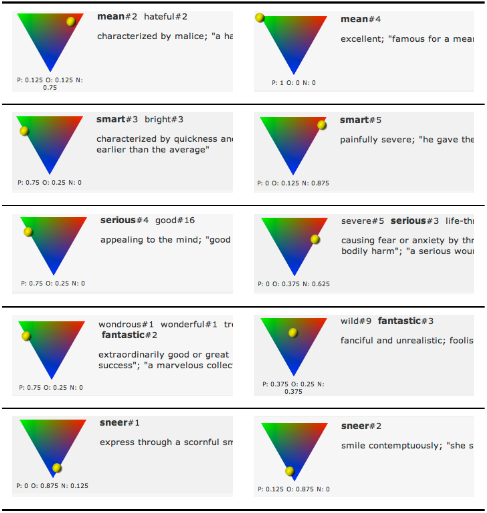
Part-of-speech (POS) tagging

Word	Tag1	Val1	Tag2	Val2
arrest	jj	Positiv	vb	Negativ
even	jj	Positiv	vb	Negativ
even	rb	Positiv	vb	Negativ
fine	jj	Positiv	nn	Negativ
fine	jj	Positiv	vb	Negativ
fine	nn	Negativ	rb	Positiv
fine	rb	Positiv	vb	Negativ
help	jj	Positiv	vbn	Negativ
help	nn	Positiv	vbn	Negativ
help	vb	Positiv	vbn	Negativ
hit	jj	Negativ	vb	Positiv
mind	nn	Positiv	vb	Negativ
order	jj	Positiv	vb	Negativ
order	nn	Positiv	vb	Negativ
pass	nn	Negativ	vb	Positiv

Table: Harvard Inquirer POS contrasts.

The limits of POS tagging

1,424 cases where a (word, tag) pair is consistent with pos. and neg. lemma-level sentiment



Word	Tag	ScoreDiff
mean	s	1.75
abject	s	1.625
benign	a	1.625
modest	s	1.625
positive	s	1.625
smart	s	1.625
solid	s	1.625
sweet	s	1.625
artful	a	1.5
clean	s	1.5
evil	n	1.5
firm	s	1.5
gross	s	1.5
iniquity	n	1.5
marvellous	s	1.5
marvelous	s	1.5
plain	s	1.5
rank	s	1.5
serious	s	1.5
sheer	s	1.5
sorry	s	1.5
stunning	s	1.5
wickedness	n	1.5
	[. . .]	
unexpectedly	r	0.25
velvet	s	0.25
vibration	n	0.25
weather-beaten	s	0.25
well-known	s	0.25
whine	v	0.25
wizard	n	0.25
wonderland	n	0.25
yawn	v	0.25

Simple negation marking

The phenomenon

1. I didn’t enjoy it.
2. I never enjoy it.
3. No one enjoys it.
4. I have yet to enjoy it.
5. I don’t think I will enjoy it.

Simple negation marking

The phenomenon

1. I didn't enjoy it.
2. I never enjoy it.
3. No one enjoys it.
4. I have yet to enjoy it.
5. I don't think I will enjoy it.

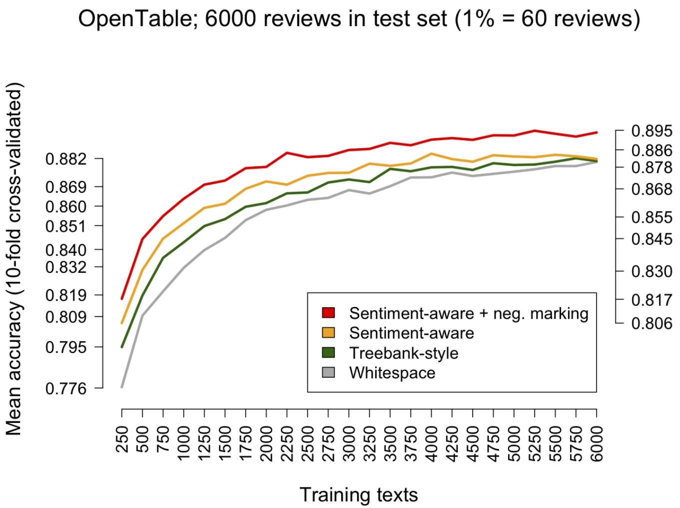
The method (Das and Chen 2001; Pang et al. 2002)

Append a _NEG suffix to every word appearing between a negation and a clause-level punctuation mark.

Simple negation marking

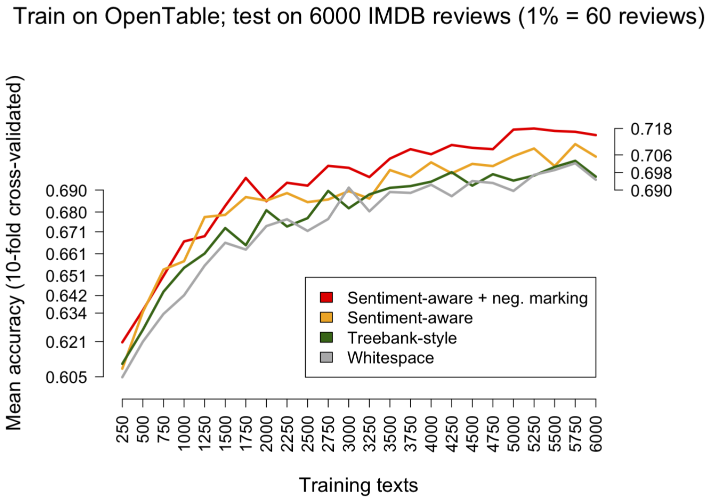
No one enjoys it.	no one_NEG enjoys_NEG it_NEG .
I don't think I will enjoy it, but I might.	i don't think_NEG i_NEG will_NEG enjoy_NEG it_NEG , but i might .

The impact of negation marking



Softmax classifier.

The impact of negation marking



Softmax classifier.

References I

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

Sanjiv Das and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 507–517, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 142–150, Portland, Oregon. Association for Computational Linguistics.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 43–52, New York, NY, USA. ACM.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, Philadelphia. Association for Computational Linguistics.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2020. DynaSent: A dynamic benchmark for sentiment analysis. *arXiv preprint arXiv:2012.15349*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA. Association for Computational Linguistics.

Robert West, Hristo S. Paskov, Jure Leskovec, and Christopher Potts. 2014. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2(2):297–310.