Linguistic annotations

CS 685, Spring 2020
Advanced Topics in Natural Language Processing
http://brenocon.com/cs685
https://people.cs.umass.edu/~brenocon/cs685_s20/

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

Announcements

- My OH: after lecture Tuesdays, including today
- Teammate search on post @5
  - https://piazza.com/class/kea8s4ktiue4mi?cid=5
- General course feedback form - also in Piazza "Resources" section.
  - https://docs.google.com/forms/d/e/1FAIpQLSfL9WvJAoyKW8crE8rKkScHQAcsw5fJqMlj_Miu4TV_s6n2Ew/viewform?usp=sf_link
- HW2 released tomorrow — annotation mini-project!
  - We're making it due after the project proposal
  - But you'll have to start it before then

Logistic Sigmoid

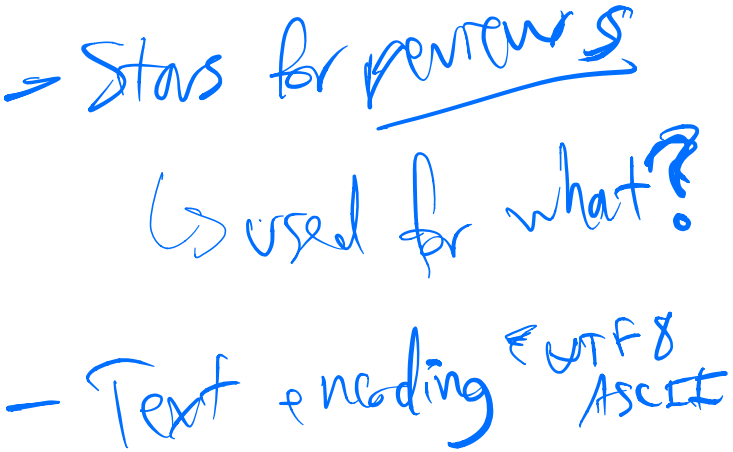
$z = \beta' x$

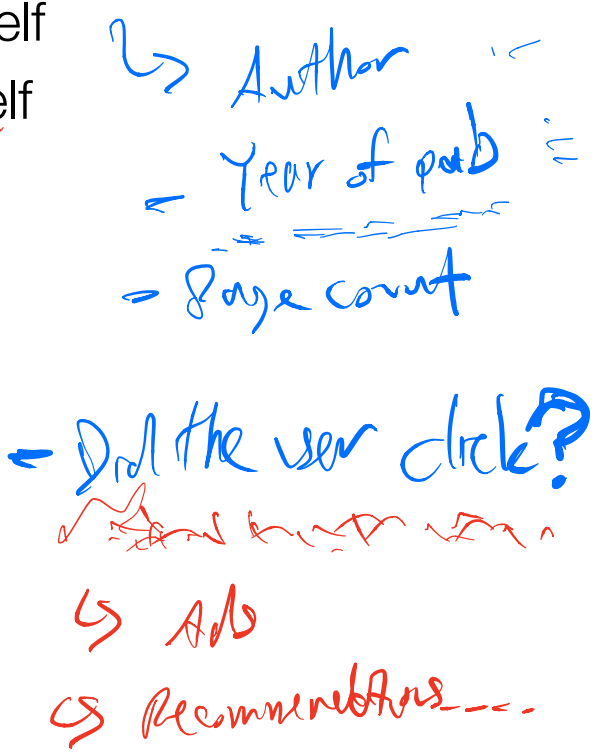
$P(y=1|x) = g(z)$

$$g(z) = \frac{e^z}{1+e^z} \frac{\frac{1}{e^z}}{\frac{1}{e^z}} = \frac{1}{1+e^{-z}}$$

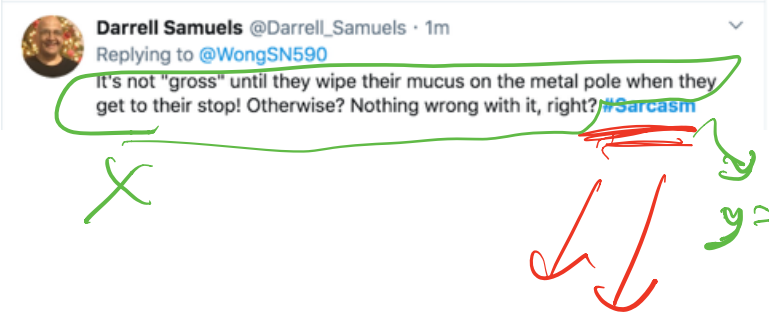
Where to get labels?

- Natural annotations
  - Metadata - information associated with text document, but not in text itself
  - Clever patterns from text itself





Sarcasm Detection



Proceedings of the Ninth International AAAI Conference on Web and Social Media

Contextualized Sarcasm Detection on Twitter

David Bamman and Noah A. Smith
School of Computer Science
Carnegie Mellon University
{dbamman,nasmith}@cs.cmu.edu

Abstract

Sarcasm requires some shared knowledge between speaker and audience; it is a profoundly *contextual* phenomenon. Most computational approaches to sarcasm detection, however, treat it as a purely linguistic matter, using information such as lexical cues and their corresponding sentiment as predictive features. We show that by including extra-linguistic information from the context of an utterance on Twitter – such as properties of the author, the audience and the immediate communicative environment – we are able to achieve gains in accuracy compared to purely linguistic features in the detection of this complex phenomenon, while also shedding light on features of interpersonal interaction that enable sarcasm in conversation.

people who know each other well than between those who do not.

In all of these cases, the relationship between author and audience is central for understanding the sarcasm phenomenon. While the notion of an "audience" is relatively well defined for face-to-face conversations between two people, it becomes more complex when multiple people are present (Bell 1984), and especially so on social media, when a user's "audience" is often unknown, underspecified or "collapsed" (boyd 2008; Marwick and boyd 2011), making it difficult to fully establish the shared ground required for sarcasm to be detected, and understood, by its intended (or imagined) audience.

We present here a series of experiments to discern the effect of extra-linguistic information on the detection of sar-

A Large Self-Annotated Corpus for Sarcasm

Mikhail Khodak and **Nikunj Saunshi** and **Kiran Vodrahalli**
Computer Science Department, Princeton University
35 Olden St., Princeton, New Jersey 08540
{mkhodak,nsaunshi,knv}@cs.princeton.edu

Abstract

We introduce the Self-Annotated Reddit Corpus (**SARC**)[1], a large corpus for sarcasm research and for training and evaluating systems for sarcasm detection. The corpus has 1.3 million sarcastic statements — 10 times more than any previous dataset — and many times more instances of non-sarcastic statements, allowing for learning in regimes of both balanced and unbalanced labels. Each statement is furthermore *self-annotated* — sarcasm is labeled by the author and not an independent annotator — and provided with user, topic, and conversation context. We evaluate the corpus for accuracy, compare it to previous related corpora, and provide baselines for the task of sarcasm detection.

1 Introduction

Sarcasm detection is an important component of many natural language processing (NLP) systems, with direct relevance to natural language understanding, dialogue systems, and text mining. However, detecting sarcasm is difficult because it occurs infrequently and is difficult for even human annotators to discern (Wallace et al., 2014). Despite these properties, existing datasets

self-annotated labels and does not consist of low-quality text snippets from Twitter[2]. With more than a million examples of sarcastic statements, each provided with author, topic, and contex information, the dataset also exceeds all previous sarcasm corpora by an order of magnitude. This dataset is possible due to the comment structure of the social media site Reddit[3] as well its frequently-used and standardized annotation for sarcasm.

Following a discussion of corpus construction and relevant statistics, in Section 4 we present results of a manual evaluation on a subsample of the data as well as a direct comparison with alternative sources. Then in Section 5 we examine simple methods of detecting sarcasm on both a balanced and unbalanced version of our dataset.

2 Related Work

Since our main contribution is a corpus and not a method for sarcasm detection, we point the reader to a recent survey by Joshi et al. (2016) that discusses many interesting efforts in this area. Note that many of the works the authors mention will be discussed by us in this section, with many papers using their own datasets; this illustrates the need for common baselines for evaluation.

Sarcasm datasets can largely be distinguished by the sources used to get sarcastic and non-sarcastic statements, the amount of human anno-

Where to get labels?

- Natural annotations
  - Metadata - information associated with text document, but not in text itself
  - Clever patterns from text itself
- New human annotations
  - Yourself
  - Your friends
  - Hire people locally
  - Hire people online
    - Mechanical Turk — most commonly used crowdsourcing site
    - (For larger/more expensive tasks: Upwork/ODesk)

Secure https://www.mturk.com/mturk/welcome

amazonmechanical turk beta
Artificial Artificial Intelligence

Your Account | HITs | Qualifications

Already have an account?
Sign in as a Worker | Requester

Introduction | **Dashboard** | **Status** | **Account Settings**

Mechanical Turk is a marketplace for work.

We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.

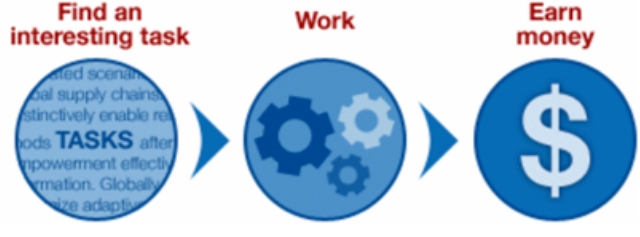
247,056 HITs available. View them now.

Make Money
by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work



Get Results
from Mechanical Turk Workers

Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. Get Started.

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Annotation process

- To pilot a new task, requires an iterative process
  - Look at data to see what's possible
  - Conceptualize the task, try it yourself
  - Write annotation guidelines
  - Have annotators try to do it. Where do they disagree? What feedback do they have?
  - Revise guidelines and repeat
- If you don't do this, your labeled data will have lots of unclear, arbitrary, and implicit decisions inside of it
- [GO TO SPREADSHEET]

Annotations quality

- Measurement theory from social sciences asks about
  - **Validity**: is it right?
  - **Reliability**: is it repeatable?

[Quinn et al. 2010]

Validity

- The annotations you got - **are they right**?
- Face validity
- Construct validity → reflect a deeper concept?
  - Convergent
  - Discriminant
- Predictive validity → predict sth else

Reliability

- The annotations you got - **are they repeatable**?
  - How much do two humans agree on labels?
    - Simple quantitative metric! Next slide.
  - Difficulty of task. Human training? Human motivation/effort?
- Goal: get the human performance "upper bound"
  - Does human agreement rate represent an upper bound for machine performance?

Measuring agreement rates

- Assume two annotators both judge a set of items
- **Agreement rate**: proportion of time two annotators agree
  - i.e., accuracy of one annotator matching the other
- Chance-adjusted agreement
  - If some classes predominate, raw agreement rate may be misleading
  - Many similar measures for this: Cohen's kappa, Krippendorff's alpha, etc.
- Cohen's kappa

$$\kappa = \frac{\text{agreement} - E[\text{agreement}]}{1 - E[\text{agreement}]}$$

- [GO TO SPREADSHEET]