Sequencing Legal DNA

NLP for Law and Political Economy

9. Language Models

Q&A Page

`bit.ly/NLP-QA09`

Outline

Analyzing polarization in social media: Method and application to tweets on 21 mass shootings

Demszky, Garg, Voigt, Zou, Gentzkow, Shapiro, and Jurafsky (2019)

- Dataset:
  - tweets about 21 mass shooting events in USA, 2015-2018.
  - $N = 10,000$ (out of 4.4 million tweets from the firehose archive).
  - Party affiliation identified off of whether account follows more Democrats or Republicans
- Text partisanship:
  - measure from Gentzkow, Shapiro, and Taddy (2019) – roughly, text distance between Democrat and Republican twitter accounts.

Sentence Embeddings for Topic Assignment

- Train GloVe embeddings on tweets and create Create Arora et al (2017) embeddings:
- Cluster the embeddings using k-means
- Identify and drop hard-to-classify tweets:
  1. compute ratio of distance to closest topic and distance to second-closest topic.
  2. drop tweets above the 75th percentile.
- Validation using Amazon Mechanical Turk to choose number of clusters:
  - Identify word intruder: five from one cluster, one from another cluster.
  - Identify tweet intruder: three from one cluster, and one from another cluster.

Topic Content

Topic	10 Nearest Stems
news (19%)	break, custodi, #breakingnew, #updat, confirm, fatal, multipl, updat, unconfirm, sever
investigation (9%)	suspect, arrest, alleg, apprehend, custodi, charg, accus, prosecutor, #break, ap
shooter's identity & ideology (11%)	extremist, radic, racist, ideolog, label, rhetor, wing, blm, islamist, christian
victims & location (4%)	bar, thousand, california, calif, among, los, southern, veteran, angel, via
laws & policy (14%)	sensibl, regul, requir, access, abid, #gunreformnow, legisl, argument, allow, #guncontolnow
solidarity (13%)	affect, senseless, ach, heart, heartbroken, sadden, faculti, pray, #prayer, deepest
remembrance (6%)	honor, memori, tuesday, candlelight, flown, vigil, gather, observ, honour, capitol
other (23%)	dude, yeah, eat, huh, gonna, ain, shit, ass, damn, guess

- The embedding method resulted in more coherent topics (better MTurk validation for words and tweets) than a topic model. $k = 8$ got best coherence.
  - Appendix reports samples of tweets for each topic (but does not say how samples were selected).

Between-topic vs within-topic polarization

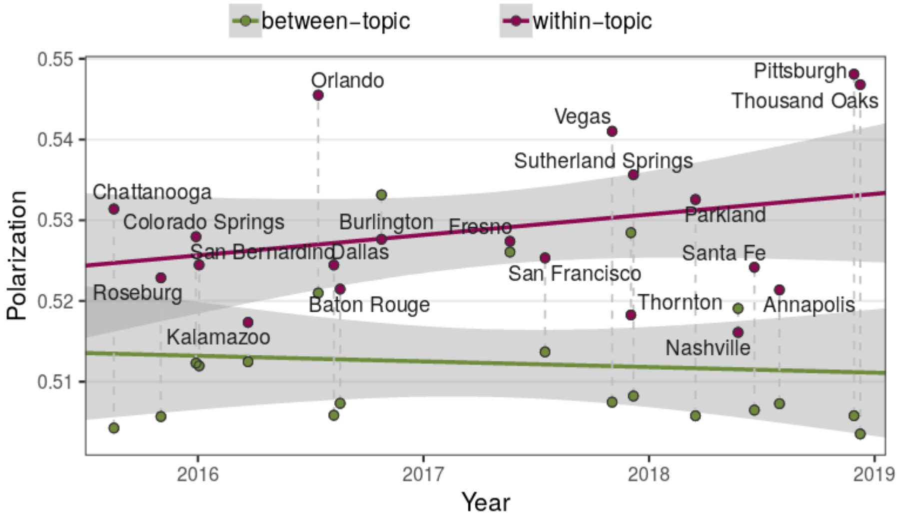
- Within-topic polarization: compute partisan text distance separately by the tweet clusters.

Between-topic vs within-topic polarization

- Within-topic polarization: compute partisan text distance separately by the tweet clusters.
- Between-topic polarization: Compute partisan text distance using cluster counts, rather than token counts.

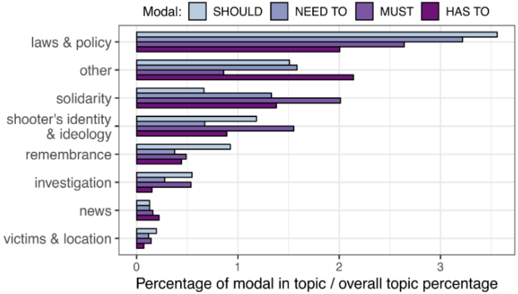
Between-topic vs within-topic polarization

- Within-topic polarization: compute partisan text distance separately by the tweet clusters.
- Between-topic polarization: Compute partisan text distance using cluster counts, rather than token counts.



Modality

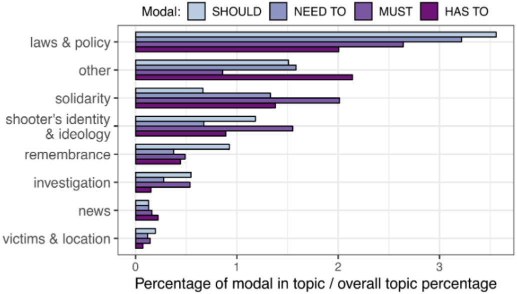
This roller coaster debate **MUST** STOP! Sensible gun ownership is one thing but assault weapons massacre innocent lives. The savagery of gore at #Parkland was beyond belief & **must** be the last.
In times of tragedy **shouldn't** we all come together?! Prayers for those harmed in the #PlannedParenthood shooting.
Communities **need to** step up and address white on white crime like the Las Vegas massacre. White men are out of control.
he BLM protest shooting, planned parenthood, now cali... domestic terrorism will crumble this country, SANE PPL **HAVE TO** FIGHT BACK
Shooting cops is horrible, cannot be condoned. But **must be** understood these incidents are outgrowth of decades of police abuses. #BatonRouge
1. Islamic terrorists are at war with us 2. Gun free zones = kill zones 3. Americans **should be** allowed to defend themselves #Chattanooga
Las Vegas shooting Walmart shooting and now 25 people killed in Texas over 90 people killed Mexico **should** build that wall to keep the US out
CNN reporting 20 dead, 42 injured in Orlando night club shooting. Just awful. The US **must** act to control guns or this carnage will continue.



- Count the four most frequent necessity modals in the data: should, must, have to, need to.
  - in this context, they are used as calls to action.

Modality

This roller coaster debate **MUST** STOP! Sensible gun ownership is one thing but assault weapons massacre innocent lives. The savagery of gore at #Parkland was beyond belief & **must** be the last.
In times of tragedy **shouldn't** we all come together?! Prayers for those harmed in the #PlannedParenthood shooting.
Communities **need to** step up and address white on white crime like the Las Vegas massacre. White men are out of control.
he BLM protest shooting, planned parenthood, now cali... domestic terrorism will crumble this country, SANE PPL **HAVE TO** FIGHT BACK
Shooting cops is horrible, cannot be condoned. But **must be** understood these incidents are outgrowth of decades of police abuses. #BatonRouge
1. Islamic terrorists are at war with us 2. Gun free zones = kill zones 3. Americans **should be** allowed to defend themselves #Chattanooga
Las Vegas shooting Walmart shooting and now 25 people killed in Texas over 90 people killed Mexico **should** build that wall to keep the US out
CNN reporting 20 dead, 42 injured in Orlando night club shooting. Just awful. The US **must** act to control guns or this carnage will continue.



- Count the four most frequent necessity modals in the data: should, must, have to, need to.
  - in this context, they are used as calls to action.
- Democrats use modals more than Republicans; Republicans seem more fatalistic.

Partisanship of Topics, by Race of Shooter

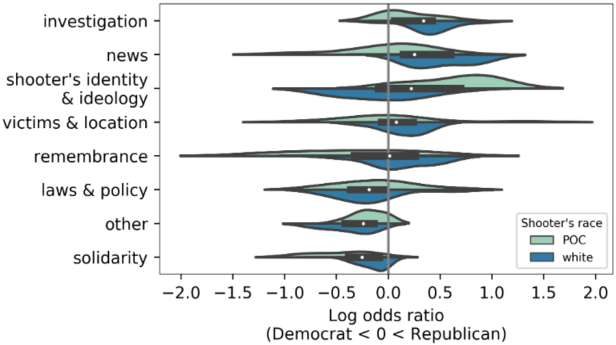


Figure 7: The plot shows the kernel density of the partisan log odds ratios of each topic (one observation per event). The white points show the median and the black rectangles the interquartile range across events.

Partisan Framing Devices: Words

- Partisanship of phrases from supervised model:

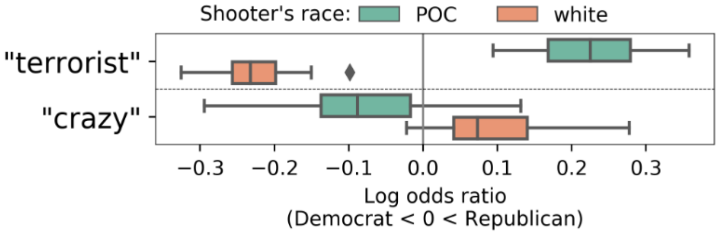


Figure 8: The log odds ratios of "terrorist" and "crazy" across events, grouped by the shooter's race. The boxes show the interquartile range and the diamond an outlier.

- Partisan valence of "terrorist" and "crazy" flip depending on race of shooter (these words have the largest racial difference in the joint vocabulary).

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).
- Domain propagation (Hamilton et al 2018):
  - pick 5-11 representative words per emotion category (Appendix E)
  - for each word in vocabulary, compute average distance to each member of each category. take 30 closest words as lexicon.

Affect (Emotions)

- Starting point: Emotion lexicon from Mohammad and Turney (2013), available at `saifmohammad.com`.
  - 14,182 words assigned to sentiment (positive/negative) and emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust).
- Domain propagation (Hamilton et al 2018):
  - pick 5-11 representative words per emotion category (Appendix E)
  - for each word in vocabulary, compute average distance to each member of each category. take 30 closest words as lexicon.

sadness senseless, loss, tragedi, lost, devast, sad, love, griev, horrif, terribl, pain, violenc, condol, broken, hurt, feel, victim, mourn, horrifi, will, grief, ach, suffer, sick, kill, aw, sicken, evil, massacr, mad

disgust disgust, sick, shame, ignor, wrong, blame, hell, ridicul, idiot, murder, evil, coward, sicken, feel, disgrac, slaughter, action, bad, insan, attack, pathet, outrag, polit, terrorist, mad, damn, lose, shit, lie, asshol

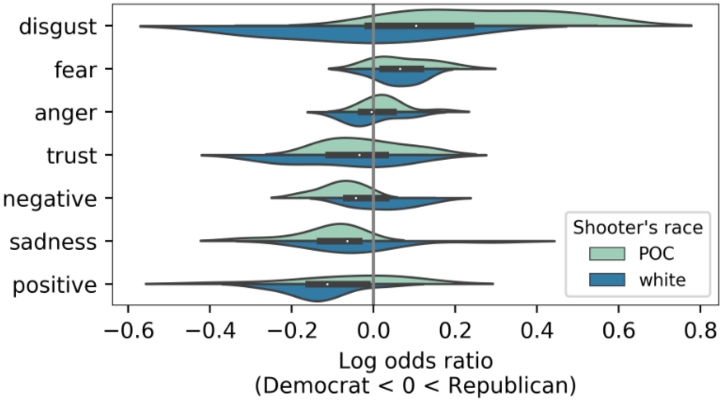
anger gun, will, murder, kill, violenc, wrong, shoot, bad, death, attack, feel, shot, action, arm, idiot, crazi, crimin, terrorist, mad, hell, crime, blame, fight, ridicul, insan, shit, die, threat, terror, hate

fear danger, threat, fear, arm, gun, still, shooter, attack, feel, fight, hide, murder, shot, shoot, bad, kill, chang, serious, violenc, forc, risk, defend, warn, govern, concern, fail, polic, wrong, case, terrorist

trust school, like, good, real, secur, show, nation, don, protect, call, teacher, help, law, great, save, true, wonder, respons, sad, answer, person, feel, safe, thought, continu, love, guard, church, fact, support

Partisanship of Affect Categories

- Compute partisanship scores using affect-category counts:



- Disgust affect flips along partisan lines depending on race of shooter.

We provide an NLP framework to uncover four linguistic dimensions of political polarization in social media: topic choice, framing, affect and illocutionary force. We quantify these aspects with existing lexical methods, and propose clustering of tweet embeddings as a means to identify salient topics for analysis across events; human evaluations show that our approach generates more cohesive topics than traditional LDA-based models. We apply our methods to study 4.4M tweets on 21 mass shootings. We provide evidence that the discussion of these events is highly polarized politically and that this polarization is primarily driven by partisan differences in framing rather than topic choice. We identify framing devices, such as grounding and the contrasting use of the terms "terrorist" and "crazy", that contribute to polarization. Results pertaining to topic choice, affect and illocutionary force suggest that Republicans focus more on the shooter and event-specific facts (news) while Democrats focus more on the victims and call for policy changes. Our work contributes to a deeper understanding of the way group divisions manifest in language and to computational methods for studying them.

1. What is the research question?
2. What is the problem solved?
3. What is being measured?
4. How does the measurement help answer the research question?

Outline

Language Modeling

- "Language Modeling" refers to the task of teaching an algorithm to predict/generate language.

Language Modeling

- "Language Modeling" refers to the task of teaching an algorithm to predict/generate language.
- The standard approach uses the Markov assumption: future words are independent of the past given the present and some finite number of previous rounds.
  - A kth order markov-assumption assumes that the next word in a sequence depends only on the last k words:

$$\Pr(w_{i+1}|w_{1:i}) \approx \Pr(w_{i+1}|w_{i-k:i})$$

- The task is to learn $\Pr(w_{i+1}|w_{1:i})$ given a large corpus.

Perplexity

- Perplexity is an information-theoretic measurement of how well a probability model predicts a sample.
- Given a text corpus of n words $\{w_1, ... w_n\}$ and a language model function $\Pr(\cdot)$, the perplexity is:

$$2^{-\frac{1}{n}\sum_{i=1}^{n} \log \widehat{\Pr}(w_i | w_{1:i-1})}$$

- Good language models (i.e., reflective of real language usage) assign high probabilities to the observed words in the corpus, resulting in lower (better) perplexity values.

N-Gram Approach to Language Modeling

- Let $\#(w_{i:j})$ be the count of the sequence of words $w_{i:j}$ in the corpus.
- The MLE estimate for the probability of a word given the previous k words is

$$\widehat{\Pr}(w_{i+1}|w_{i-k:i}) = \frac{\#(w_{i-k:i+1})}{\#(w_{i-k:i})}$$

N-Gram Approach to Language Modeling

- Let $\#(w_{i:j})$ be the count of the sequence of words $w_{i:j}$ in the corpus.
- The MLE estimate for the probability of a word given the previous k words is

$$\widehat{\Pr}(w_{i+1}|w_{i-k:i}) = \frac{\#(w_{i-k:i+1})}{\#(w_{i-k:i})}$$

- The obvious problem:
  - if $w_{i-k:i+1}$ was never observed in the corpus, $\widehat{\Pr}$ is zero.
  - zero events are quite common because many phrases are unique.

Neural Language Modeling (Goldberg 2017)

- Input:
  - preceding sequence (context words) $w_{1:k}$.
  - V is a finite vocabulary, including special symbols for unknown words, start of sentence, and end of sentence.
  - Each context word is associated with an embedding vector.
  - The input vector $\boldsymbol{x}$ is a concatenation of the word vectors.
- Output:
  - probability distribution over the next word.

Neural Language Modeling (Goldberg 2017)

- Input:
  - preceding sequence (context words) $w_{1:k}$.
  - V is a finite vocabulary, including special symbols for unknown words, start of sentence, and end of sentence.
  - Each context word is associated with an embedding vector.
  - The input vector $\boldsymbol{x}$ is a concatenation of the word vectors.
- Output:
  - probability distribution over the next word.
- Model architecture could be an MLP applied to the embeddings, a CNN, an RNN, or a transformer.
- Computational cost of these language models is the softmax across the vocabulary in the final layer, which becomes slower with an increase in vocabulary size.

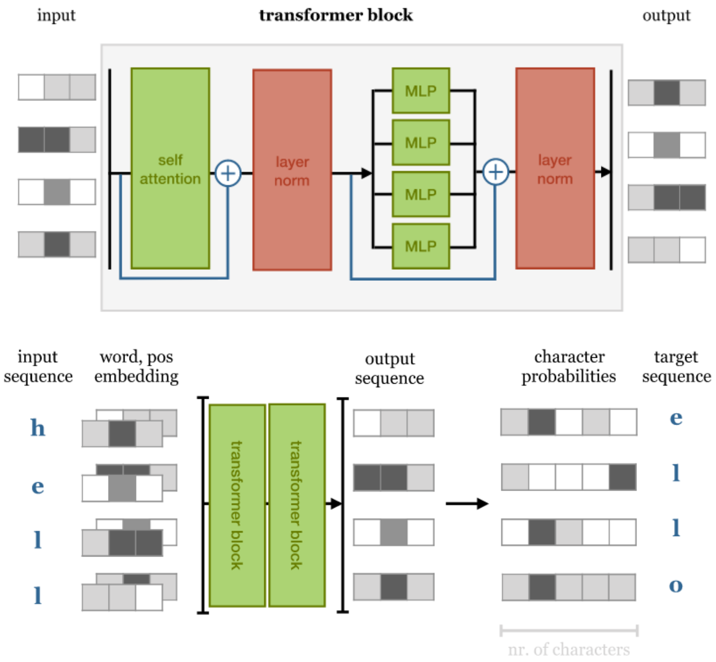
Autoregressive vs Autoencoding Language Models

- **Autoregressive models** (e.g. GPT):
  - pretrained on classic language modeling task: guess the next token having read all the previous ones.
  - during training, attention heads only view previous tokens, not subsequent tokens.
  - ideal for text generation.
- **Autoencoding models** (e.g. BERT):
  - pretrained by dropping/shuffling input tokens and trying to reconstruct the original sequence.
  - usually build bidirectional representations and get access to the full sequence.
  - can be fine-tuned and achieve great results on many tasks, e.g. text classification.

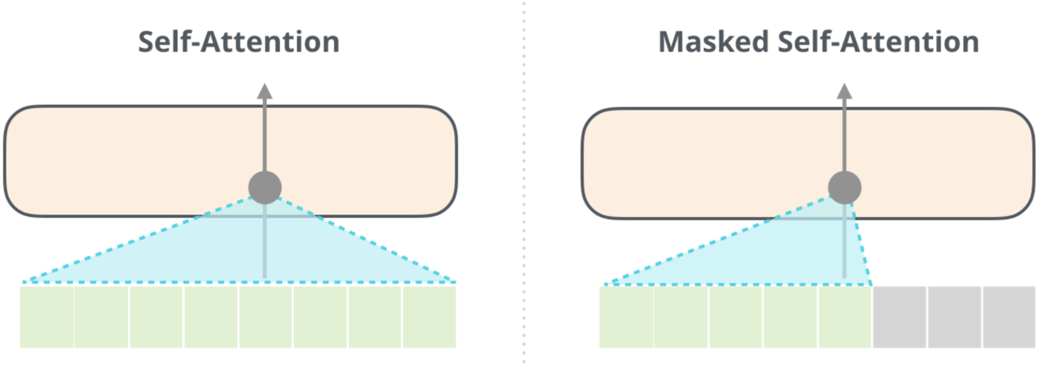
Outline

- Demzsky et al (2019)
- Language Modeling
- Autoregressive Transformers (e.g. GPT)
  - GPT
  - Conditioned Generation
  - Applications
- Autoencoding Transformers (e.g. BERT)
- Some Technical Details
- Causal BERT

Text generation transformer



Masked Self-Attention



- An autoregressive model's attention mechanism only looks at previous tokens.

Outline

GPT = Generative Pre-Trained Transformer

GPT = Generative Pre-Trained Transformer

- GPT-1: the first autoregressive transformer model (2018)
  - trained on the Books corpus.
  - train on a language modeling task, as well as a multi-task that adds a supervised learning task.

GPT = Generative Pre-Trained Transformer

- GPT-1: the first autoregressive transformer model (2018)
  - trained on the Books corpus.
  - train on a language modeling task, as well as a multi-task that adds a supervised learning task.
- GPT-2 (2019):
  - all articles linked from Redditt with at least 3 upvotes (8 million documents, 40 GB of text)
  - dispense with supervised learning task, make some other architectural adjustments
  - make model much bigger

GPT = Generative Pre-Trained Transformer

- GPT-1: the first autoregressive transformer model (2018)
  - trained on the Books corpus.
  - train on a language modeling task, as well as a multi-task that adds a supervised learning task.
- GPT-2 (2019):
  - all articles linked from Redditt with at least 3 upvotes (8 million documents, 40 GB of text)
  - dispense with supervised learning task, make some other architectural adjustments
  - make model much bigger
- GPT-3 (2020):
  - use an even bigger corpus (Common Crawl, WebText2, Books1, Books2 and Wikipedia)
  - make model much, much bigger

OPENAI'S NEW MULTITALENTED AI WRITES, TRANSLATES, AND SLANDERS

A step forward in AI text-generation that also spells trouble

By James Vincent | Feb 14, 2019, 12:00pm EST

Howard, co-founder of Fast.AI agrees. "I've been trying to warn people about this for a while," he says. "We have the technology to totally fill Twitter, email, and the web up with reasonable-sounding, context-appropriate prose, which would drown out all other speech and be impossible to filter."

What can GPT-2 and GPT-3 Do?

- State-of-the-art perplexity on diverse corpora.

What can GPT-2 and GPT-3 Do?

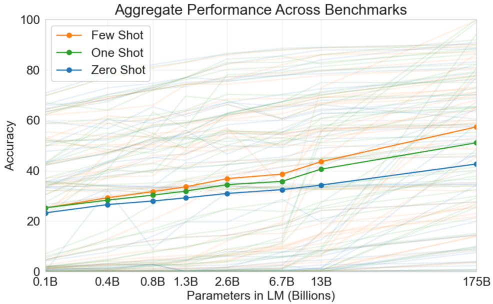
- State-of-the-art perplexity on diverse corpora.
- Reading Comprehension: *<context> <question> A:*
- Summarization: `TL;DR:`
- Question Answering: `A:`

What can GPT-2 and GPT-3 Do?

- State-of-the-art perplexity on diverse corpora.
- Reading Comprehension: *<context> <question> A:*
- Summarization: `TL;DR:`
- Question Answering: `A:`
- Translation:
  - *[English sentence 1] = <French sentence 1>*
  - *[English sentence 2] = <French sentence 2>*
  - *.....*
  - *[Source sentence] =*

GPT Model Sizes

- GPT-1:
  - 768-dimensional word embeddings
  - 12 transformer blocks with 12 attention heads
  - 512-token context window
  - $\approx$ 117M parameters
- GPT-2:
  - 1600-dimensional word embeddings
  - 48 blocks with 48 attention heads
  - 1024-token context window
  - $\approx$ 1.5B parameters
- GPT-3:
  - 12,888-dimensional word embeddings
  - 96 blocks with 96 attention heads
  - 2048-token context window
  - $\approx$ 175B parameters



Story Generation (GPT-2)

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

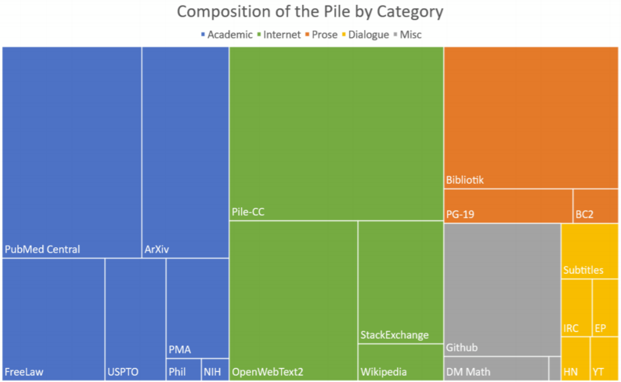
Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

- GPT-3 is even better: `https://beta.openai.com/playground`

GPT-Neo

- The Eleuther AI team is building an open-source equivalent of GPT-3.
  - `https://github.com/EleutherAI/gpt-neo/`
- Dataset: The Pile:
  - 825GB of text comprising 22 high-quality datasets (Gao et al 2020)



- Have already trained and released 1.3B and 2.7B parameter models based on GPT architecture.

Outline

Conditioned Generation

- Text generators can use metadata, for example on the speaker.
  - e.g., Li et al (2016) learn a categorical embedding for each user who wrote a response, in order to produce automated responses in the style of each user.
- As a side effect of training the generator, the network learns user embeddings, producing similar vectors to users who have similar communication styles.
  - At test time, one can influence the style of the generated response by feeding in a particular user (or average user vector) as a conditioning context.

Grover: Modify GPT-2 for conditioned fake news generation

- Next predicted word is based not only on previous words (text body), but also on metadata:

$$p(\text{domain}, \text{date}, \text{authors}, \text{headline}, \text{body}).$$

  - e.g., user specifies domain, date, and headline. Grover generates a) text body, then b) authors, then c) more realistic headline.

Grover: Modify GPT-2 for conditioned fake news generation

- Next predicted word is based not only on previous words (text body), but also on metadata:

$$p(\text{domain}, \text{date}, \text{authors}, \text{headline}, \text{body}).$$

  - e.g., user specifies domain, date, and headline. Grover generates a) text body, then b) authors, then c) more realistic headline.
- fields are dropped with probability 10%, and all but body with probability 35% $\rightarrow$ Grover learns to perform unconditional generation.
- RealNews Corpus:
  - Authors scraped all news articles from the 5000 news domains from Google News, Dec 2016 to April 2019
  - 120 GB of uncompressed text after deduplication

Grover: Results

- Grover can generate articles in particular newspaper styles.
  - metadata is important: it reduced perplexity from 9.3 to 8.7

Grover: Results

- Grover can generate articles in particular newspaper styles.
  - metadata is important: it reduced perplexity from 9.3 to 8.7
- Grover is better than human generated fake news:

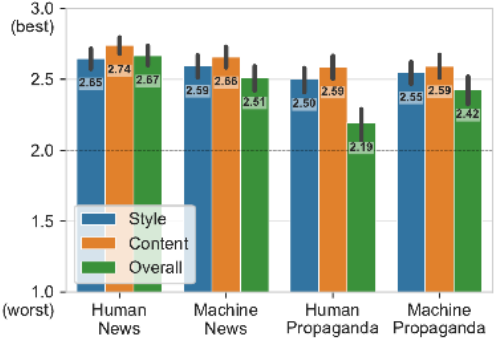


Figure 4: Human evaluation. For each article, three annotators evaluated style, content, and the overall trustworthiness; 100 articles of each category were used. The results show that propaganda generated by GROVER is rated more plausible than the original human-written propaganda.

Grover: Fake News Detection

- They take Grover's document embedding for the true article and generated articles, and feed that to a true/false classifier.
  - they do it in the test set month (April 2019); training ended in March 2019.
- Grover can identify its own fake articles with 81% accuracy.
  - better than other models including BERT

Plug and Play Language Model (PPLM)

- Setup:
  - Let $p(w)$ = unconditional language model (LM), a probability distribution for word w given history w_0

Plug and Play Language Model (PPLM)

- Setup:
  - Let $p(w)$ = unconditional language model (LM), a probability distribution for word w given history w_0
  - $p(w|a)$ = conditional LM, generating sentences with attribute a.

Plug and Play Language Model (PPLM)

- Setup:
  - Let $p(w)$ = unconditional language model (LM), a probability distribution for word w given history w_0
  - $p(w|a)$ = conditional LM, generating sentences with attribute a.
- By Bayes rule:

$$p(w|a) \propto p(a|w)p(w)$$

  - can be approximated efficiently using the method in Ngyuen et al (2016).

Plug and Play Language Model (PPLM)

- Setup:
  - Let $p(w)$ = unconditional language model (LM), a probability distribution for word w given history w_0
  - $p(w|a)$ = conditional LM, generating sentences with attribute a.
- By Bayes rule:

$$p(w|a) \propto p(a|w)p(w)$$

  - can be approximated efficiently using the method in Ngyuen et al (2016).
- PPLM insight:
  - instead of training $p(w|a)$ from scratch, take pre-trained $p(w)$, learn auxiliary model $p(a|w)$, and approximate $p(w|a)$ using Bayes rule.
- At each word step, use the gradients from the language model and the auxiliary model to increase both probabilities.
  - works to maintain fluency of generated language.

Outline

Kreps et al: News Generation Experiment

- Kreps et al (2019) evaluate the use of GPT-2 for fake news generation.
- Experiment:
  - New York Times story on North Korea.
  - GPT-2 gets 2 sentences, then generates 20 short news stories.
  - Researchers manually selected the most credible out of these twenty.

Kreps et al: News Generation Experiment

- Kreps et al (2019) evaluate the use of GPT-2 for fake news generation.
- Experiment:
  - New York Times story on North Korea.
  - GPT-2 gets 2 sentences, then generates 20 short news stories.
  - Researchers manually selected the most credible out of these twenty.
  - Respondents rank stories as credible or not.
- Results:
  - For the larger GPT-2 models, machine-generated articles were rated the same as the true article.

Peric, Mijic, Stammbach, and Ash (2020)

"Legal Language Modeling with Transformers"

- we show that GPT models can generate believable legal text
  - law students cant distinguish real from generated snippets
  - the model embeddings can be used to classify real vs fake snippets

Peric, Mijic, Stammbach, and Ash (2020)

"Legal Language Modeling with Transformers"

- we show that GPT models can generate believable legal text
  - law students cant distinguish real from generated snippets
  - the model embeddings can be used to classify real vs fake snippets
- Try it out:

  `https://53478.gradio.app/`

- we are working on conditioned generation, e.g. to generate dissenting opinions in response to majority opinions.

Outline

Demzsky et al (2019)

Language Modeling

Autoregressive Transformers (e.g. GPT)
- GPT
- Conditioned Generation
- Applications

Autoencoding Transformers (e.g. BERT)

Some Technical Details

Causal BERT

BERT (and RoBERTa)

- BERT = Bidirectional Encoder Representations from Transformers
  - RoBERTa = Robust BERT
- Architecture:
  - a stack of transformer blocks with a self-attention layer and an MLP.
  - The largest BERT model has 24 blocks, embedding dimension of 1024, and 16 attention heads.
    $\approx$ 340M parameters to learn.

BERT (and RoBERTa)

- BERT = Bidirectional Encoder Representations from Transformers
  - RoBERTa = Robust BERT
- Architecture:
  - a stack of transformer blocks with a self-attention layer and an MLP.
  - The largest BERT model has 24 blocks, embedding dimension of 1024, and 16 attention heads.
    $\approx$ 340M parameters to learn.
- Task: Masked language modeling:
  - 15% of words masked
  - if masked: replace with [MASK] 80% of the time, a random token 10% of the time, and left unchanged 10% of the time.
  - model has to predict the original word.

BERT (and RoBERTa)

- BERT = Bidirectional Encoder Representations from Transformers
  - RoBERTa = Robust BERT
- Architecture:
  - a stack of transformer blocks with a self-attention layer and an MLP.
  - The largest BERT model has 24 blocks, embedding dimension of 1024, and 16 attention heads.
    $\approx$ 340M parameters to learn.
- Task: Masked language modeling:
  - 15% of words masked
  - if masked: replace with [MASK] 80% of the time, a random token 10% of the time, and left unchanged 10% of the time.
  - model has to predict the original word.
- Unlike GPT, BERT attention observes all tokens in the sequence, reads backwards and forwards (bidirectional).

BERT (and RoBERTa)

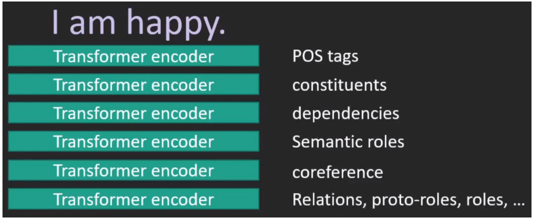
- BERT = Bidirectional Encoder Representations from Transformers
  - RoBERTa = Robust BERT
- Architecture:
  - a stack of transformer blocks with a self-attention layer and an MLP.
  - The largest BERT model has 24 blocks, embedding dimension of 1024, and 16 attention heads.
    $\approx$ 340M parameters to learn.
- Task: Masked language modeling:
  - 15% of words masked
  - if masked: replace with [MASK] 80% of the time, a random token 10% of the time, and left unchanged 10% of the time.
  - model has to predict the original word.
- Unlike GPT, BERT attention observes all tokens in the sequence, reads backwards and forwards (bidirectional).
- Corpus:
  - 800M words from English books (modern work, from unpublished authors), by Zhu et al (2015).
  - 2.5B words of text from English Wikipedia articles (without markup).

- BERT obtains state-of-the-art results on many NLP tasks (see Devlin et al 2019).
- The model can be fine-tuned as needed.
  - e.g. for sentiment analysis, text classification.

- BERT obtains state-of-the-art results on many NLP tasks (see Devlin et al 2019).
- The model can be fine-tuned as needed.
  - e.g. for sentiment analysis, text classification.

BERT Rediscovers the Classical NLP Pipeline

Ian Tenney[1] **Dipanjan Das**[1] **Ellie Pavlick**[1,2]
[1]Google Research [2]Brown University
{iftenney,dipanjand,epavlick}@google.com



- Like the earlier and later layers in ELMo, the earlier and later layers in BERT respectively encode more functional and more semantic information.

Model Distillation

- Large transformer models such as BERT can be compressed.
  - a smaller model is given the inputs and BERT's outputs as the label.
  - works almost as well (97% of full BERT performance) and 60% faster

$\rightarrow$ When using pre-trained models, usually better to use DistilBERT or DistilGPT.

Model Distillation

- Large transformer models such as BERT can be compressed.
  - a smaller model is given the inputs and BERT's outputs as the label.
  - works almost as well (97% of full BERT performance) and 60% faster

$\rightarrow$ When using pre-trained models, usually better to use DistilBERT or DistilGPT.

- one reason this works:
  - for a given masked token, the student model observes probabilities across the whole vocabulary, not just the single true token.

Extracting morality dimension from BERT

Extracting morality dimension from BERT

- Schramowski et al (2019) project BERT embeddings onto a “moral subspace”, analogous to what Bolukbasi et al (2016) do for a gender subspace:

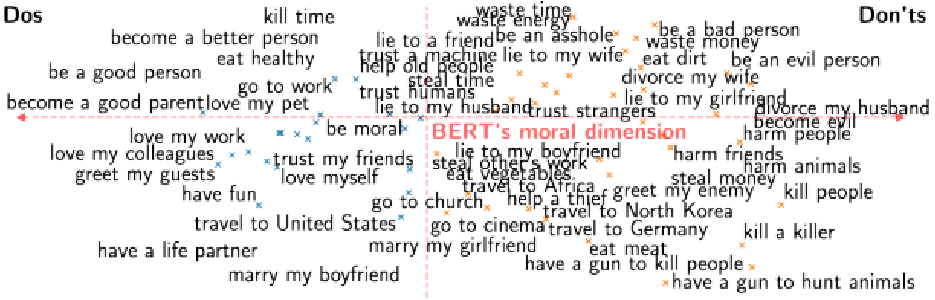


Figure 4: Context-based actions projected —based on PCA computed by selected atomic actions— along two axes: x (top PC) defines the moral direction m (Left: *Dos* and right: *Don'ts*). Compare Tab. 9(Appendix) for detailed moral bias scores.

- tricky to distinguish morality from sentiment, though.

Outline

Subword Tokenization

Modern transformer models use subword tokenization:

- construct character-level n-grams
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

Subword Tokenization

Modern transformer models use subword tokenization:

- construct character-level n-grams
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

e.g., BERT's SentencePiece tokenizer:

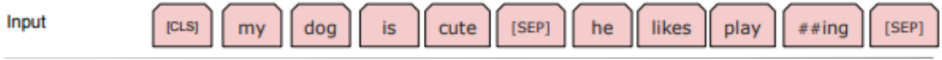
- character-level byte-pair encoder, learns character n-grams to breaks words like "playing" into "play" and "##ing".
- have to fix a vocabulary size: e.g. BERT uses 30K.

Subword Tokenization

Modern transformer models use subword tokenization:

- construct character-level n-grams
- whitespace treated the same as letters
- all letters to lowercase, but add a special character for the next letter being capitalized.

e.g., BERT's SentencePiece tokenizer:



- character-level byte-pair encoder, learns character n-grams to breaks words like "playing" into "play" and "##ing".
- have to fix a vocabulary size: e.g. BERT uses 30K.

This approach works well because the softmax over the vocabulary for output layer is computationally expensive.

- preferred to a character-level model because some sequences are relatively frequent – e.g., the sequence "the" shows up more often than the letter "Z".

More on Position Embeddings

- Last time we looked at categorical positional index embeddings (this is the standard).
  - alternatives include non-linear transformations of the index, e.g. sine / cosine (as done in the original Vaswani et al 2017 transformer paper).
- Transformer-XL Approach:
  - for a given sequence, model observes the final encoding vector of the previous sequence
  - Position vectors encode not the absolute position, but the distance to the current output.

More on Position Embeddings

- Last time we looked at categorical positional index embeddings (this is the standard).
  - alternatives include non-linear transformations of the index, e.g. sine / cosine (as done in the original Vaswani et al 2017 transformer paper).
- Transformer-XL Approach:
  - for a given sequence, model observes the final encoding vector of the previous sequence
  - Position vectors encode not the absolute position, but the distance to the current output.
- Some recent research shows that positional encodings (or any direct information on word order) are often not necessary after all (Irie et al 2019; Schlag et al 2021, Sinha et al 2021).

Scaled Dot Product Self-Attention

- Recall from the Week 8 lecture, transformers consist of attention mechanisms:

$$h_i = \sum_{j=1}^{n_L} a(x_i, x_j) x_j$$

Scaled Dot Product Self-Attention

- Recall from the Week 8 lecture, transformers consist of attention mechanisms:

$$h_i = \sum_{j=1}^{n_L} a(x_i, x_j) x_j$$

- The specification for general self-attention used in more recent transformers is

$$a(x_i, x_j)x_j = \text{softmax}\Big(\frac{\overbrace{(W_Q x_i)^\top}^{\text{"query"}} \overbrace{(W_K x_j)}^{\text{"key"}}}{\underbrace{\sqrt{n_E}}_{\text{scaling factor}}}\Big) \underbrace{W_V x_j}_{\text{"value"}}$$

- W_Q, W_K, and W_V are the "query", "key", and "value" matrices
  - these are $n_W \times n_E$ and contain learnable model parameters.

Scaled Dot Product Self-Attention

- Recall from the Week 8 lecture, transformers consist of attention mechanisms:

$$h_i = \sum_{j=1}^{n_L} a(x_i, x_j) x_j$$

- The specification for general self-attention used in more recent transformers is

$$a(x_i, x_j)x_j = \text{softmax}\Big(\frac{\overbrace{(W_Q x_i)}^{\text{"query"}\,\top} \overbrace{(W_K x_j)}^{\text{"key"}}}{\underbrace{\sqrt{n_E}}_{\text{scaling factor}}}\Big) \underbrace{W_V x_j}_{\text{"value"}}$$

- W_Q, W_K, and W_V are the “query”, “key”, and “value” matrices
  - these are $n_W \times n_E$ and contain learnable model parameters.
- general attention is a **differentiable soft dictionary lookup**:
  - for the **query** at i, look up the similarity to each **key** j in the sequence
  - if similarity is high, weight up the associated **value** at j.

Multi-Head Attention

- Recall that convolutional neural nets use a team of filters at a given convolutional layer to specialize in learning different features.

Multi-Head Attention

- Recall that convolutional neural nets use a team of filters at a given convolutional layer to specialize in learning different features.

$$a(x_i, x_j)x_j = \text{softmax}(\frac{(W_Q^l x_i)^\top (W_K^l x_j)}{\sqrt{n_E}}) W_V^l x_j$$

- With transformers, imagine that the query-key-value matrices (W_Q^l, W_K^l, W_V^l) define one of a team of attention "heads" (analogous to "filter"), indexed by $l \in \{1, ..., n_H\}$.
  - e.g., the larger BERT model learns $n_H =$16 parallel attention heads.
  - parameters are initialized randomly, so heads will specialize in different features of sequences during training.

Multi-Head Attention

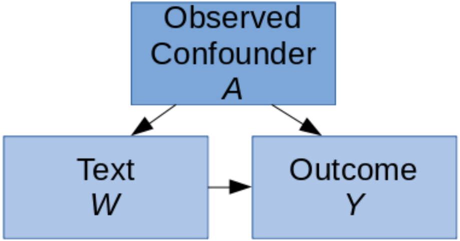
- Recall that convolutional neural nets use a team of filters at a given convolutional layer to specialize in learning different features.

$$a(x_i, x_j)x_j = \text{softmax}(\frac{(W_Q^l x_i)^\top (W_K^l x_j)}{\sqrt{n_E}}) W_V^l x_j$$

- With transformers, imagine that the query-key-value matrices (W_Q^l, W_K^l, W_V^l) define one of a team of attention "heads" (analogous to "filter"), indexed by $l \in \{1, ..., n_H\}$.
  - e.g., the larger BERT model learns n_H =16 parallel attention heads.
  - parameters are initialized randomly, so heads will specialize in different features of sequences during training.
- standard setting for n_W (from $n_W \times n_E$ attention weight matrices W_Q, W_K, W_V) is $n_W = n_E / n_H$.
- In a given transformer block:
  1. the n_W-vectors produced by each of the n_H heads are concatened
  2. the resulting $n_W n_H$-vector is encoded by another learnable parameter matrix W_O down to an n_E-vector for input to the MLP layers.

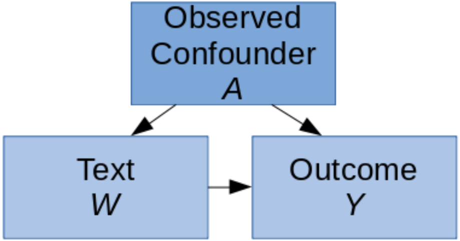
Outline

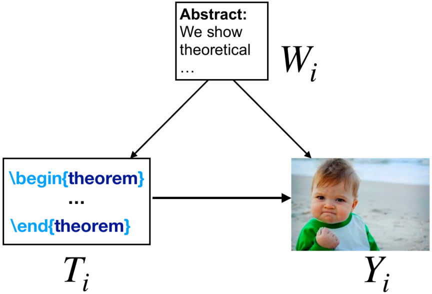
Sridhar, Veitch, and Blei (2019): Adjusting for Confounders with BERT



- This paper analyzes the problem of the effect of text features on outcomes, where the unobserved confounders are other features of the document.

Sridhar, Veitch, and Blei (2019): Adjusting for Confounders with BERT





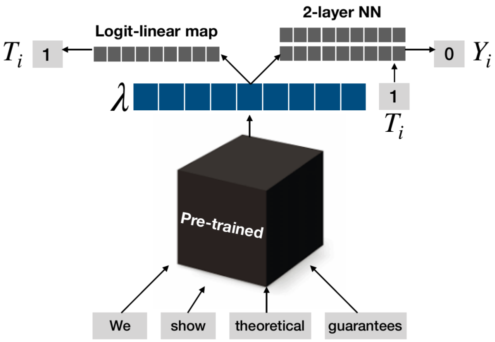
- This paper analyzes the problem of the effect of text features on outcomes, where the unobserved confounders are other features of the document.
- For example, the effect of putting a theorem in your paper on acceptance to a conference/journal.
- This paper is another example of controlling for observed confounders, but in high dimensions.

Approach

- Insight: the confounding part of the text is that which carries information about both treatment and outcome.

Approach

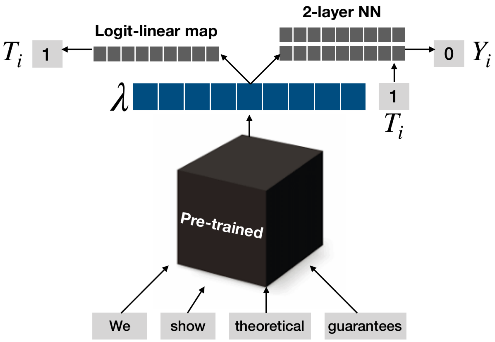
- Insight: the confounding part of the text is that which carries information about both treatment and outcome.



- Start with pre-trained BERT embeddings.
- fine-tune them on an additional multitask objective.

Approach

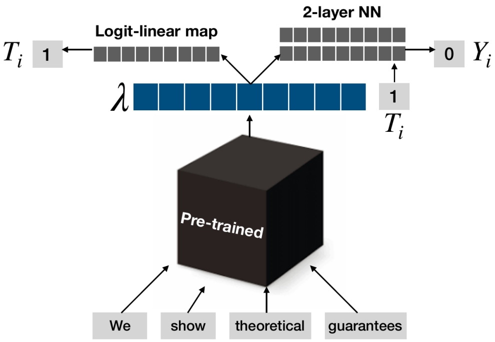
- Insight: the confounding part of the text is that which carries information about both treatment and outcome.



- Start with pre-trained BERT embeddings.
- fine-tune them on an additional multitask objective.
- 1) predict propensity score (probability of treatment given other text features.
- 2) predict outcomes conditional on treatment.

Approach

- Insight: the confounding part of the text is that which carries information about both treatment and outcome.



- Start with pre-trained BERT embeddings.
- fine-tune them on an additional multitask objective.
- 1) predict propensity score (probability of treatment given other text features.
- 2) predict outcomes conditional on treatment.
- the resulting embeddings serve as a sufficient statistic for the unobserved confounders.

Veitch et al

- The applications in the paper are not that interesting/convincing.
- Nice opportunity for an RE03 replication exercise.