Text-as-data and Content Analysis

Indira Sen

28.02.24

Universität Konstanz

Why?

- Content analysis:
    - “any technique for making inferences by objectively and systematically identifying specified characteristics of messages” [Holsti, 1969]
    - “a systematic, replicable technique for compressing many words of text into fewer content categories based on explicit rules of coding” [Berelson, 1952; GAO, 1996; Krippendorff, 1980; and Weber, 1990]

Why?

- Content analysis:
  - “any technique for making inferences by objectively and systematically identifying specified characteristics of messages” [Holsti, 1969]
  - “a systematic, replicable technique for compressing many words of text into fewer content categories based on explicit rules of coding” [Berelson, 1952; GAO, 1996; Krippendorff, 1980; and Weber, 1990]
- Questions content analysis can help us answer:
  - How does news media represent the immigration crisis?
  - What are topics that lead to arguments in long-term relationships?
  - How do citizens perceive the performance of politicians during the pandemic?

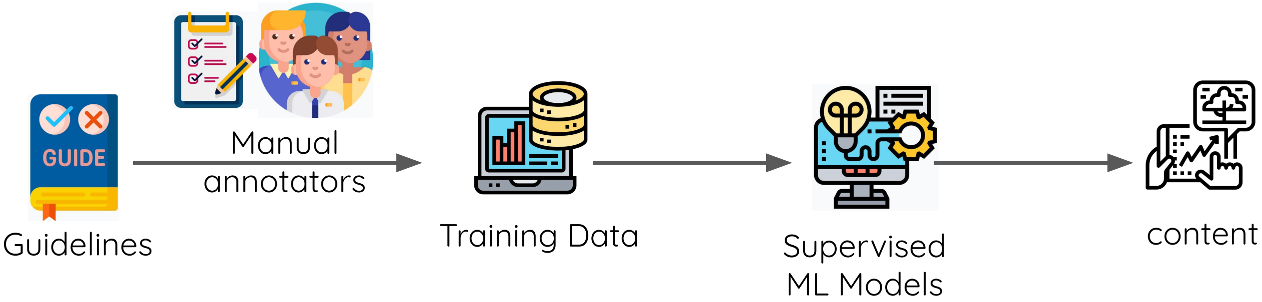
Why?

- Content analysis:
    - “any technique for making inferences by objectively and systematically identifying specified characteristics of messages” [Holsti, 1969]
    - “a systematic, replicable technique for compressing many words of text into fewer content categories based on explicit rules of coding” [Berelson, 1952; GAO, 1996; Krippendorff, 1980; and Weber, 1990]
- Questions content analysis can help us answer:
    - How does news media represent the immigration crisis?
      NYtimes articles frames [Mendelsohn’21]
    - What are topics that lead to arguments in long-term relationships?
      r/relationshipadvice posts topics
    - How do citizens perceive the performance of politicians during the pandemic?
      Twitter posts stance

Content Analysis Pipeline using ML/NLP

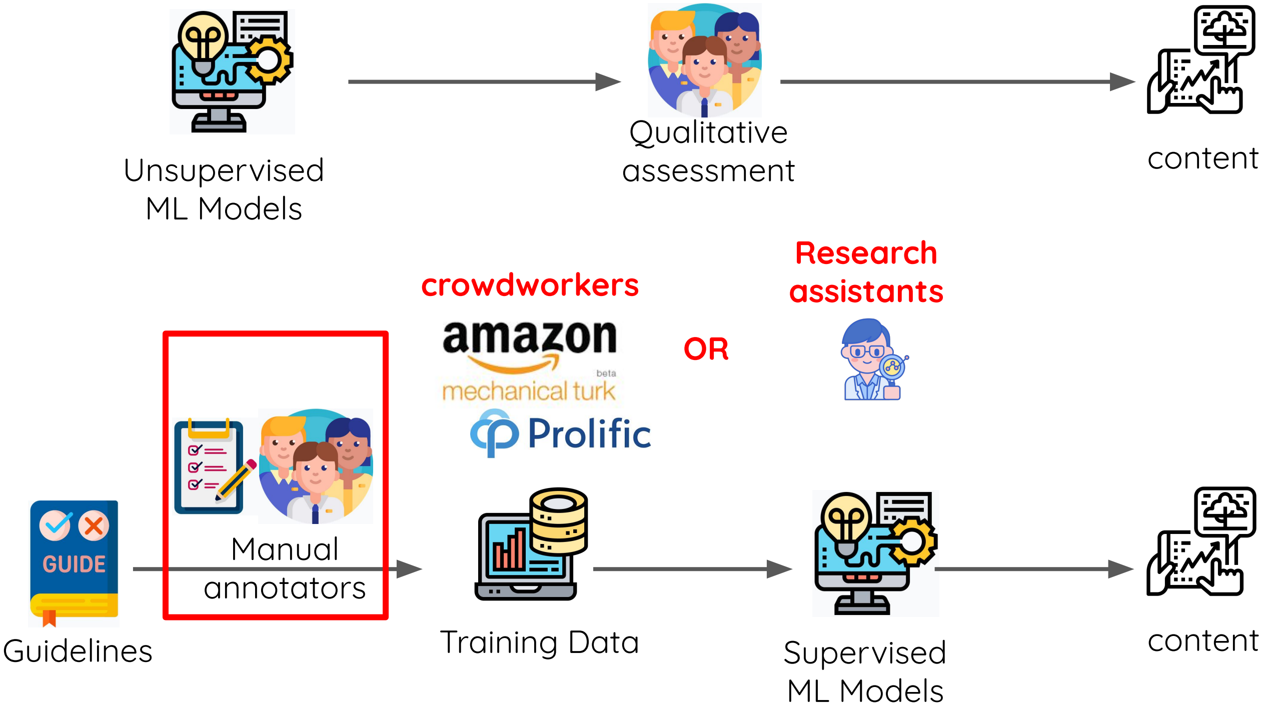
- unsupervised

- supervised



Content Analysis Pipeline using ML/NLP

- unsupervised
- supervised



After LLMs

Can Large Language Models Transform Computational Social Science?

Caleb Ziems* **William Held*** **Omar Sh**
Zhehao Zhang* **Diyi Ya**
Georgia Institute of Technology, Shanghai Jiao Tong
{cziems, wheld3, jiaaochen}@gatech.edu, zzh12138@sjtu.edu.

BRIEF REPORT | POLITICAL SCIENCES |

ChatGPT outperforms crowd workers for text-annotation tasks

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli Authors Info & Affiliations

Edited by Mary Waters, Harvard University, Cambridge, MA; received March 27, 2023; accepted June 2, 2023

i.org/10.1073/pnas.2305016120

OPEN-SOURCE LARGE LANGUAGE MODELS OUTPERFORM CROWD WORKERS AND APPROACH CHATGPT IN TEXT-ANNOTATION TASKS

After LLMs

Data \ Model	Baselines		FLAN-T5					FLAN	Chat	text-001				text-002	text-003
	Rand	Finetune	Small	Base	Large	XL	XXL	UL2	ChatGPT	Ada	Babb.	Curie	Dav.	Davinci	Davinci
Utterance Level Tasks															
Dialect	4.5	41.5	1.9	2.3	15.8	16.5	22.6	23.7	15.0	5.3	5.6	6.0	10.9	10.5	16.9
Emotion	16.7	91.7	23.9	65.3	69.1	65.9	66.7	70.3	46.2	44.6	16.1	18.7	19.3	39.8	36.5
Figurative	25.0	94.4	23.6	29.0	25.4	40.2	56.0	64.0	50.2	25.0	24.4	25.0	28.8	52.0	60.6
Humor	50.0	73.1	52.0	51.8	56.2	59.0	50.6	58.8	55.4	55.2	59.0	58.6	50.4	51.4	51.0
Ideology	33.3	61.9	33.1	39.2	48.6	49.2	54.4	48.2	54.8	–	33.3	33.3	34.3	57.6	48.2
Impl. Hate	14.3	69.9	17.7	22.7	17.9	36.3	34.5	35.9	29.7	17.1	18.6	15.7	21.3	22.7	27.1
Misinfo	50.0	82.3	50.0	55.4	69.2	70.2	71.2	77.6	69.0	–	50.4	52.2	52.6	75.6	75.0
Persuasion	12.5	40.4	14.3	19.8	43.9	43.4	†51.6	49.4	40.9	–	16.5	17.0	18.8	26.3	26.3
Sem. Chng.	50.0	65.7	50.3	50.0	†66.9	55.5	51.2	53.7	56.1	50.0	50.5	54.3	39.5	45.9	50.0
Stance	33.3	47.0	34.7	47.8	51.3	52.6	55.9	55.4	†72.0	–	33.1	31.0	48.0	57.4	41.3
Conversation Level Tasks															
Discourse	14.3	47.5	14.7	26.4	37.2	44.3	†52.5	41.9	44.5	13.1	16.5	14.3	17.0	39.8	37.8
Empathy				33.3	35.1	33.7	36.8	†39.8	37.6	–	33.1	35.3	33.3	33.3	33.3
				55.3	†57.1	53.0	53.5	53.2	52.9	50.2	50.0	50.0	50.0	50.8	55.9
				44.2	53.0	59.2	54.2	52.8	50.8	33.1	33.1	32.1	42.2	55.6	47.8
				47.2	50.4	56.8	58.8	60.8	61.6	–	52.2	50.6	49.6	50.5	57.0
				50.6	49.4	54.2	50.0	56.6	53.0	44.6	50.6	49.0	50.8	52.2	51.2
Document Level Tasks															
				–	–	–	-	–	22.3	–	–	8.6	8.6	21.6	22.9
				7.0	1.0	10.9	41.8	50.6	51.3	29.8	47.3	47.4	44.4	48.8	52.4
				34.1	34.1	32.1	49.6	40.3	58.8	32.9	35.1	33.6	25.6	48.7	44.0
Tropes	1.4	0.8	0.9	4.4	8.8	7.9	10.5	16.7	25.4	4.3	7.0	9.6	10.5	18.4	18.4

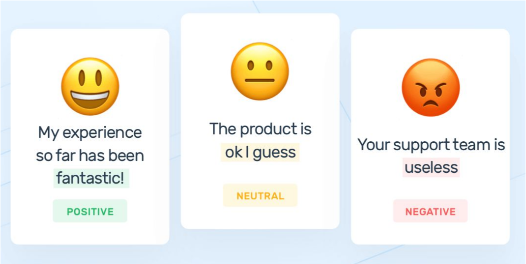
Table 2: **Zero-shot Classification Results** across our selected CSS benchmark tasks. All tasks are evaluated with accuracy, except for Event Arg. and Event Detection, which use F-1. Models which did not always follow instructions are marked with a dash. Best zero-shot models are in green; zero-shot models that are not significantly worse ($P > .05$; Paired Bootstrap test (Dror et al., 2018)) are marked blue; and † denote cases where zero-shot LLMs match or beat finetuned baselines.

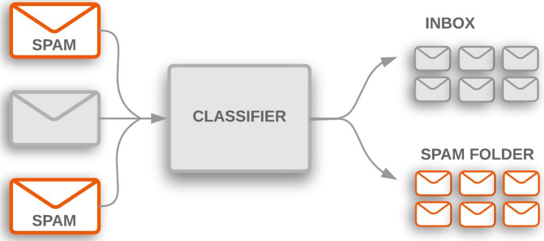
Can Large Language Models Transform Computational Social Science?

Caleb Ziems* **William Held*** **Omar Shaikh*** **Jiaao Chen***
Zhehao Zhang* **Diyi Yang***
Georgia Institute of Technology, Shanghai Jiao Tong University, Stanford University
{cziems, wheld3, jiaaochen}@gatech.edu, zzh12138@sjtu.edu.cn, {oshaikh, diyiy}@stanford.edu

Text Classification

- One of the most basic and popular NLP tasks
- Many connections to content analysis
- Supervised Machine Learning





Check out https://lena-voita.github.io/nlp_course/text_classification.html for a deeper dive into text classification

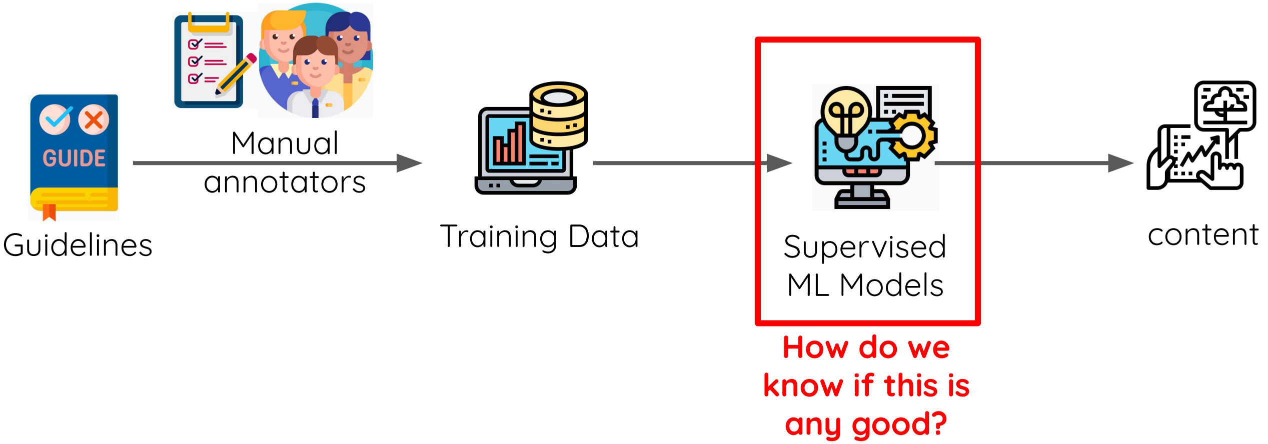
Text **Classification**

- Given ‘something’ (a document, a number, a set of numbers, etc), classify it based on a **fixed** number of categories (‘classes’)
- Numerical things are easy to classify because computers know bits (0s and 1s)
- Therefore, we turn words to numbers
    - Obtain a **representation**
    - Classify

representation → classifier

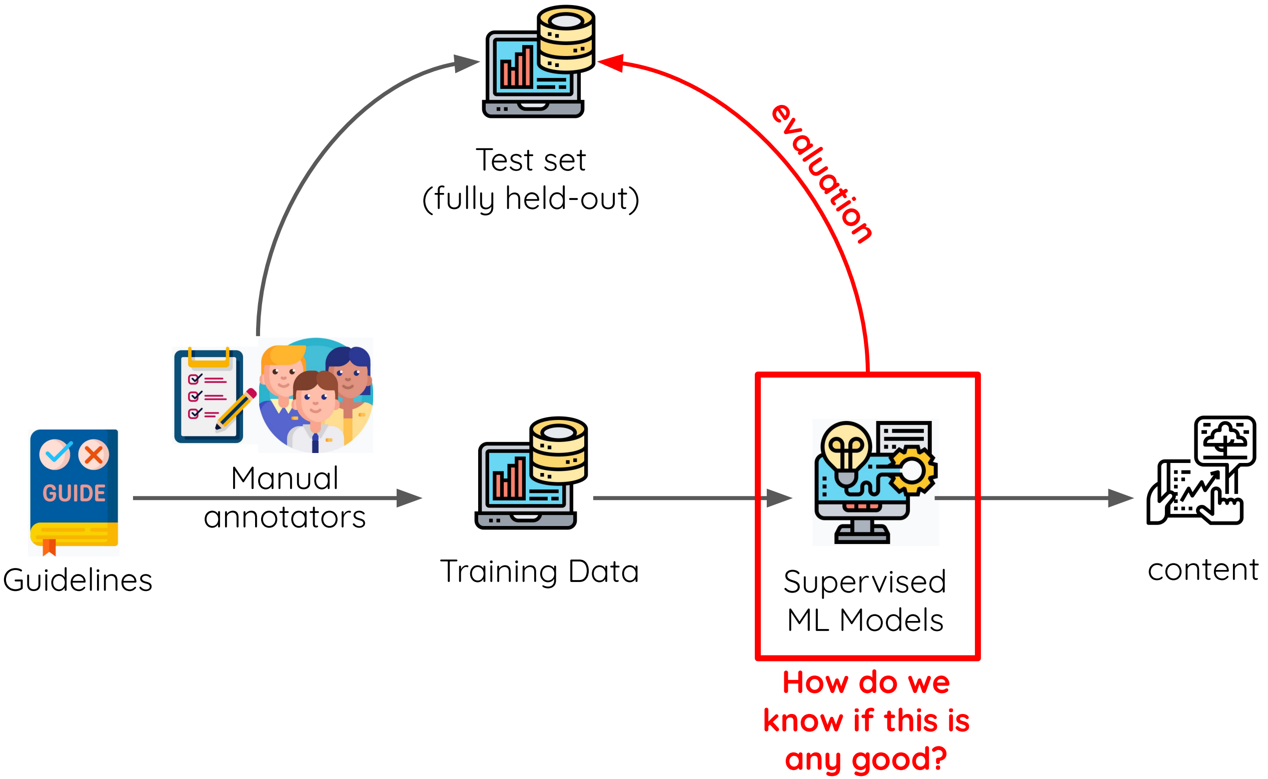
Modern Text Classification Pipeline

- supervised



Modern Text Classification Pipeline

- supervised



Evaluating Text Classification

Sources: [4][5][6][7][8][9][10][11][12] view · talk · edit

		Predicted condition			
	Total population $= P + N$	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP), hit**	**False negative (FN), type II error, miss, underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP), type I error, false alarm, overestimation**	**True negative (TN), correct rejection**	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2 TP}{2 TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

https://en.wikipedia.org/wiki/Precision_and_recall

Evaluating Text Classification

Sources: [4][5][6][7][8][9][10][11][12] view · talk · edit

		Predicted condition			
	Total population $= P + N$	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP), hit**	**False negative (FN), type II error, miss, underestimation**	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate $= \frac{FN}{P} = 1 - TPR$
	Negative (N)	**False positive (FP), type I error, false alarm, overestimation**	**True negative (TN), correct rejection**	False positive rate (FPR), probability of false alarm, fall-out $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{PN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{PP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{PN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2 PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

https://en.wikipedia.org/wiki/Precision_and_recall

Sentiment Analysis

Sentiment Analysis

Sentiment Analysis is the task of automatically annotating the sentiment / polarity of a piece of content.

- Traditionally lexica-based
- Traditionally methods were built using reviews of movies or items
- Currently several sophisticated **Machine Learning** Methods exist

Sentiment Analysis

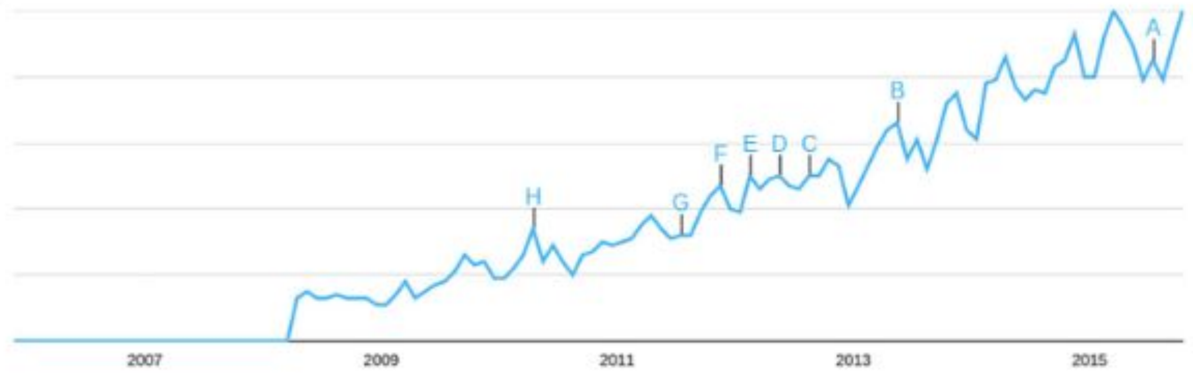
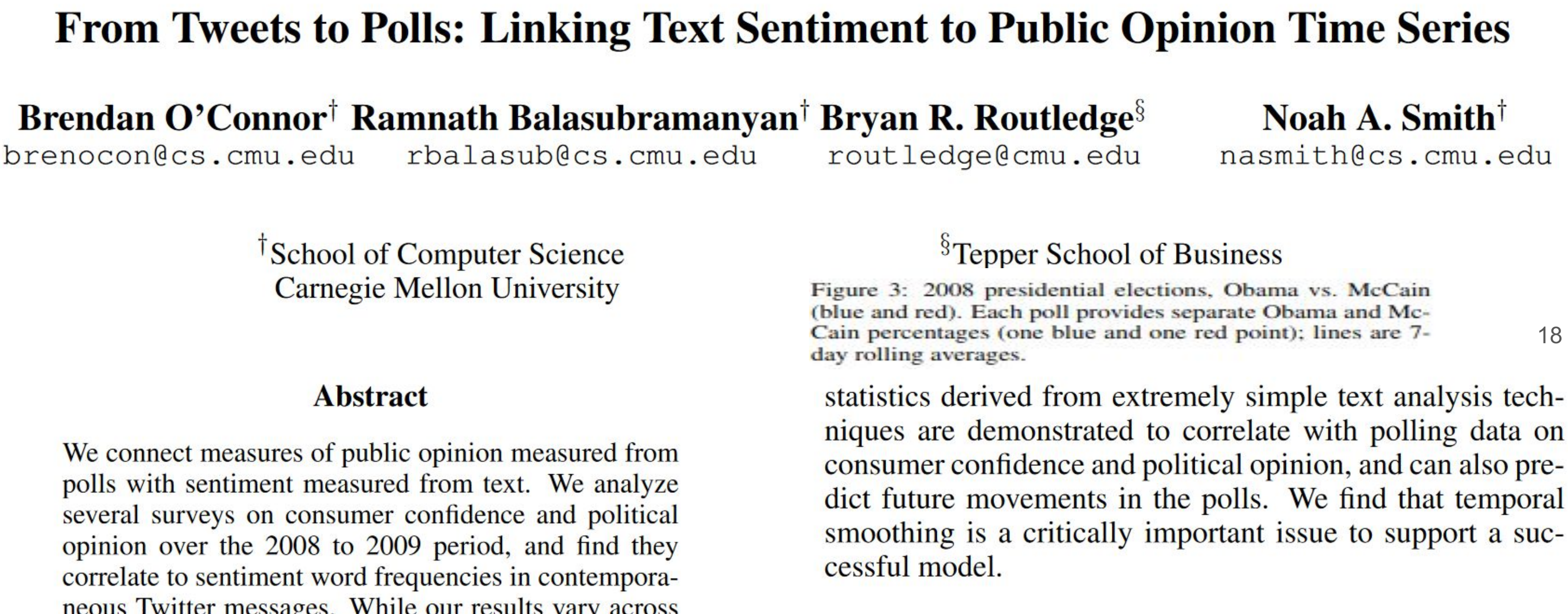


Figure 1 Searches on Google for the Query: 'Sentiment Analysis'. This figure shows the steady growth on the number of searches on the topic, according to Google Trends, mainly after the popularization of online social networks (OSNs).

Sentiment Analysis


From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series

Brendan O'Connor[†] **Ramnath Balasubramanyan**[†] **Bryan R. Routledge**[§] **Noah A. Smith**[†]
brenocon@cs.cmu.edu rbalasub@cs.cmu.edu routledge@cmu.edu nasmith@cs.cmu.edu

[†]School of Computer Science
Carnegie Mellon University

[§]Tepper School of Business

Figure 3: 2008 presidential elections, Obama vs. McCain (blue and red). Each poll provides separate Obama and McCain percentages (one blue and one red point); lines are 7-day rolling averages.

Abstract

We connect measures of public opinion measured from polls with sentiment measured from text. We analyze several surveys on consumer confidence and political opinion over the 2008 to 2009 period, and find they correlate to sentiment word frequencies in contemporaneous Twitter messages. While our results vary across

statistics derived from extremely simple text analysis techniques are demonstrated to correlate with polling data on consumer confidence and political opinion, and can also predict future movements in the polls. We find that temporal smoothing is a critically important issue to support a successful model.

Sentiment Analysis

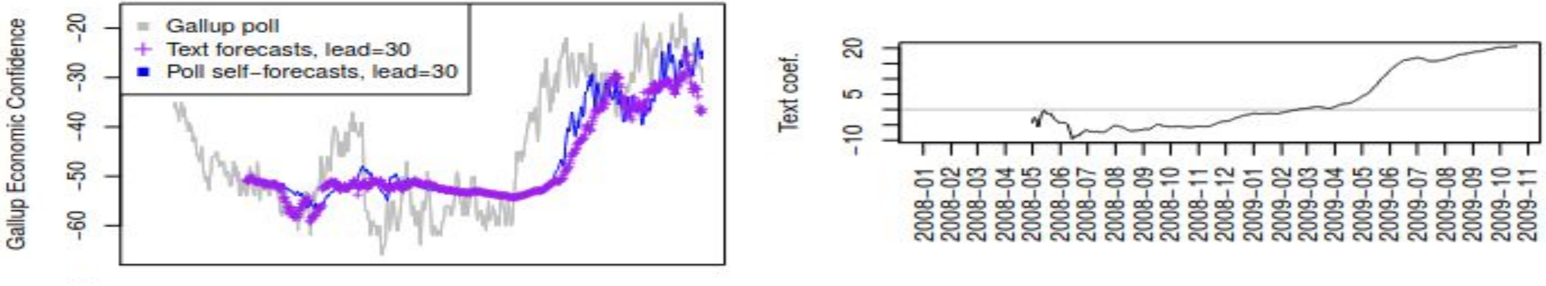


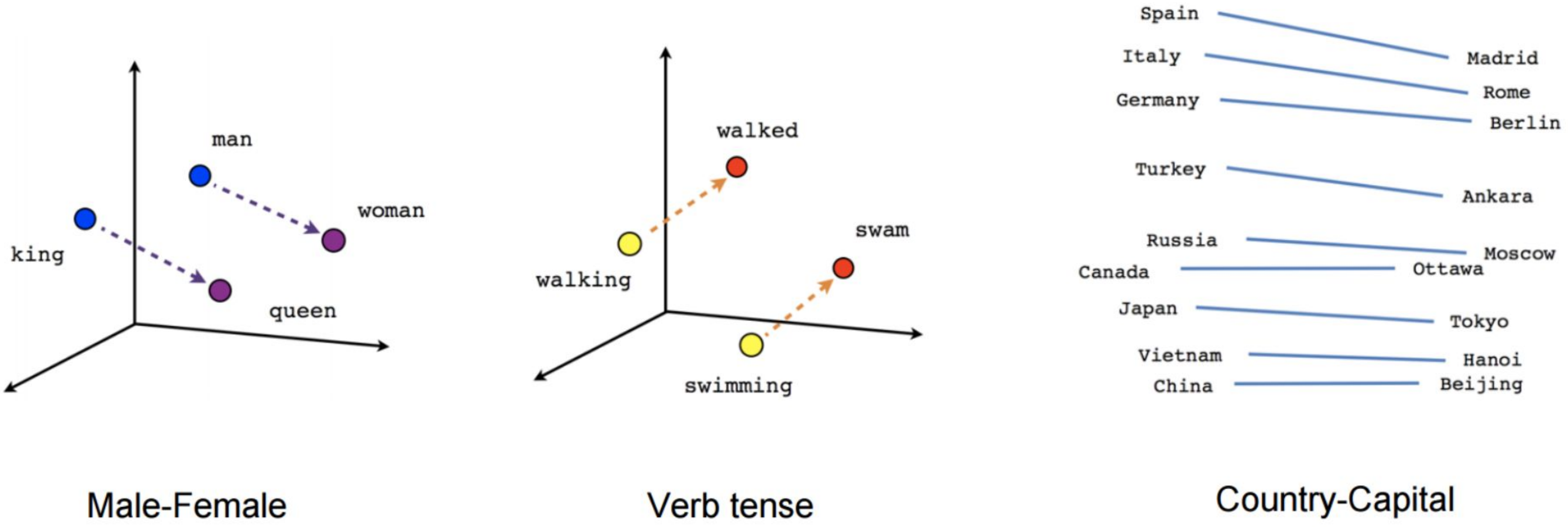
Figure 8: Rolling text-based forecasts (above), and the text sentiment (MA_t) coefficients a for each of the text forecasting models over time (below).

Sentiment Analysis

- Can be as simple as counting the positive and negative words and normalize by number of words in a text
- usually computed on a range:
  - -1 (negative) to 1 (positive)
  - very negative, negative, neutral / none, positive, positive

Let’s try it ourselves

Vector Semantics



Relations captured by word2vec. (Source: NLP's ImageNet Moment)

Vector Semantics

> “ As Wittgenstein says, ‘the meaning of words lies in their use.’ The day-to-day practice of playing language games recognizes customs and rules. It follows that a text in such established usage may contain sentences such as ‘Don’t be such an ass!’, ‘You silly ass!’, ‘What an ass he is!’ In these examples, the word ass is in familiar and habitual company, commonly collocated with you silly—, he is a silly—, don’t be such an—. **You shall know a word by the company it keeps!** ”

- John Rupert Firth, “A Synopsis of Linguistic Theory” 1957

Learn Representations of Words

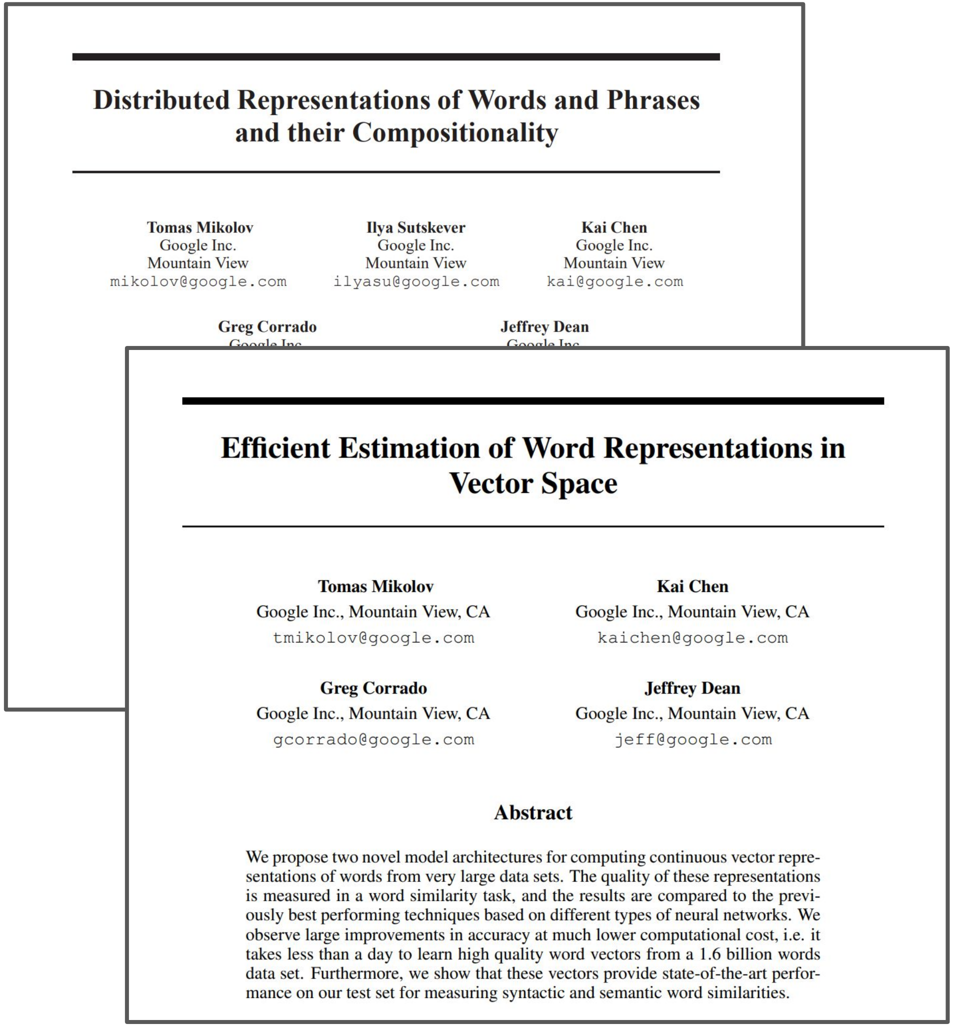
- So far: all methods based on counting; no learning or prediction

- Now: Learn low-dimensional representations (‘vectors’) through prediction: **using context to predict words in a surrounding window**

Word2Vec

- **Learned parameters:** word vectors
- **Goal:** make each vector “know” about the contexts of its word
- **How:** train vectors to predict possible contexts from words (or, alternatively, words from contexts)

Important: These are word type vectors



Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov, Google Inc., Mountain View, mikolov@google.com
Ilya Sutskever, Google Inc., Mountain View, ilyasu@google.com
Kai Chen, Google Inc., Mountain View, kai@google.com
Greg Corrado
Jeffrey Dean

Efficient Estimation of Word Representations in Vector Space

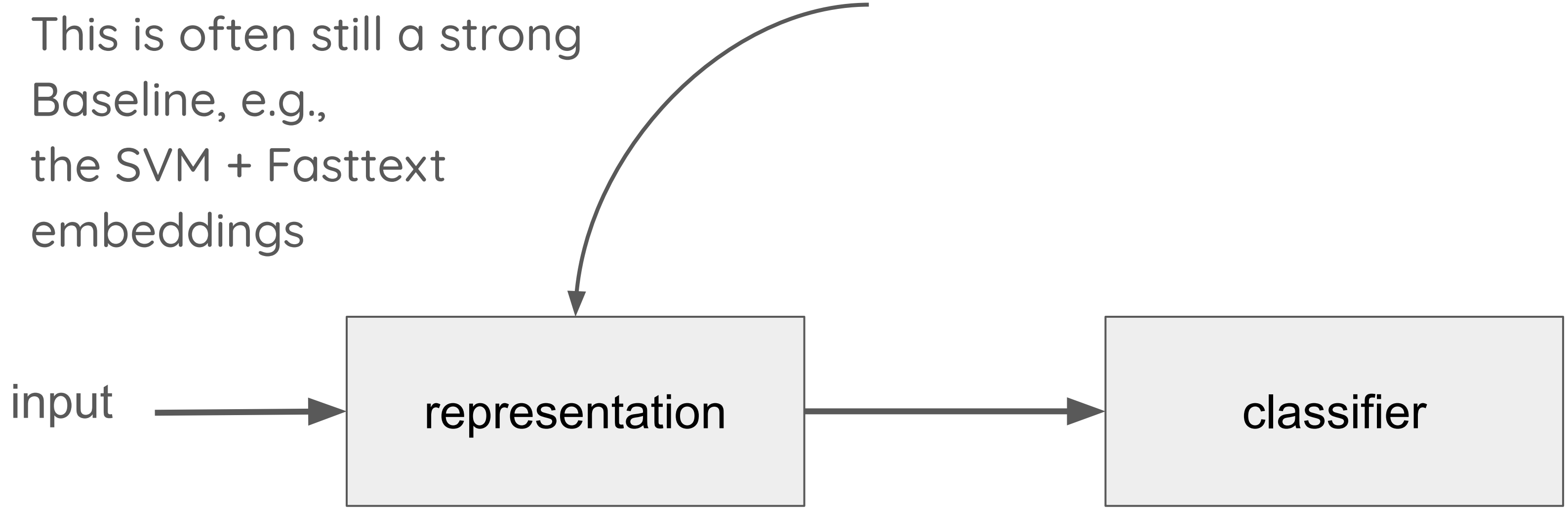
Tomas Mikolov, Google Inc., Mountain View, CA, tmikolov@google.com
Kai Chen, Google Inc., Mountain View, CA, kaichen@google.com
Greg Corrado, Google Inc., Mountain View, CA, gcorrado@google.com
Jeffrey Dean, Google Inc., Mountain View, CA, jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

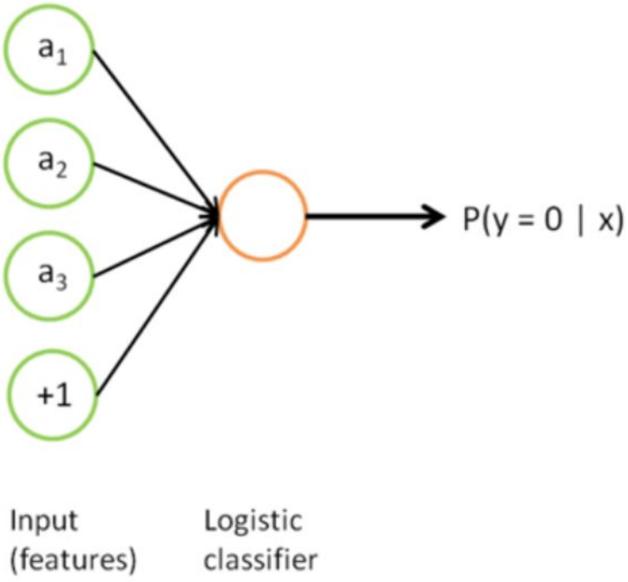
What can you do with these word vectors?

- Besides many applications like document search, word similarity, you can also use them as features for your classifier!
- This is often still a strong Baseline, e.g., the SVM + Fasttext embeddings

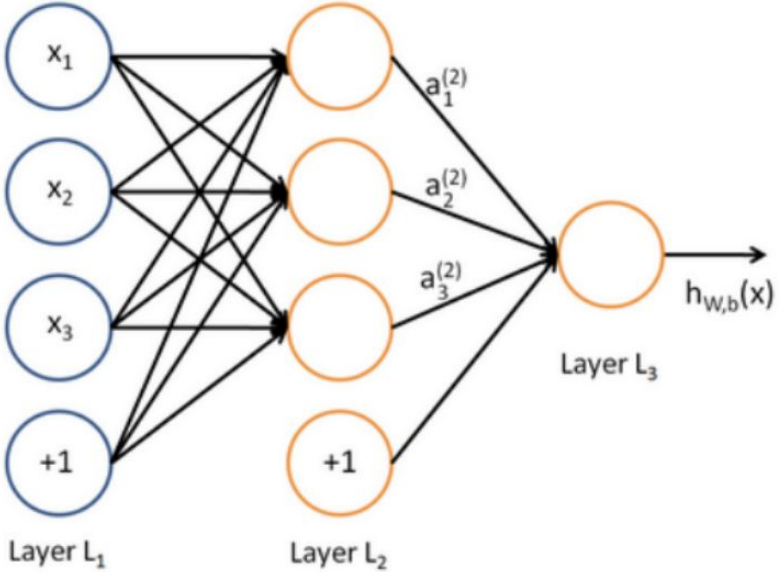


Let’s try it ourselves

Detour: Neural Networks and Deep Learning



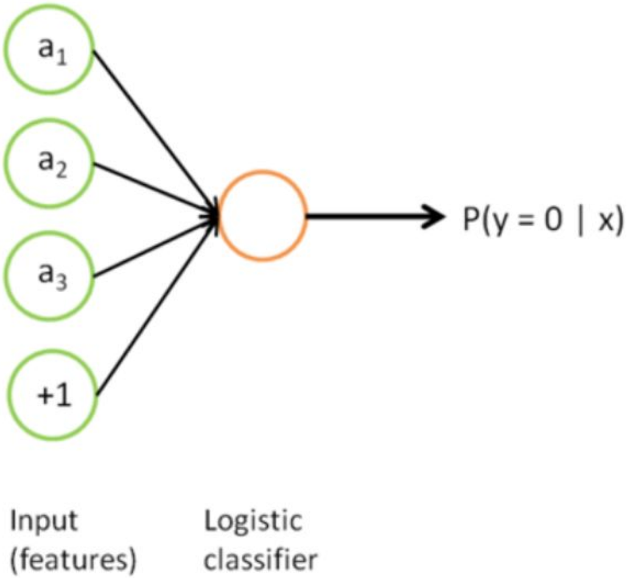
Logistic Regression



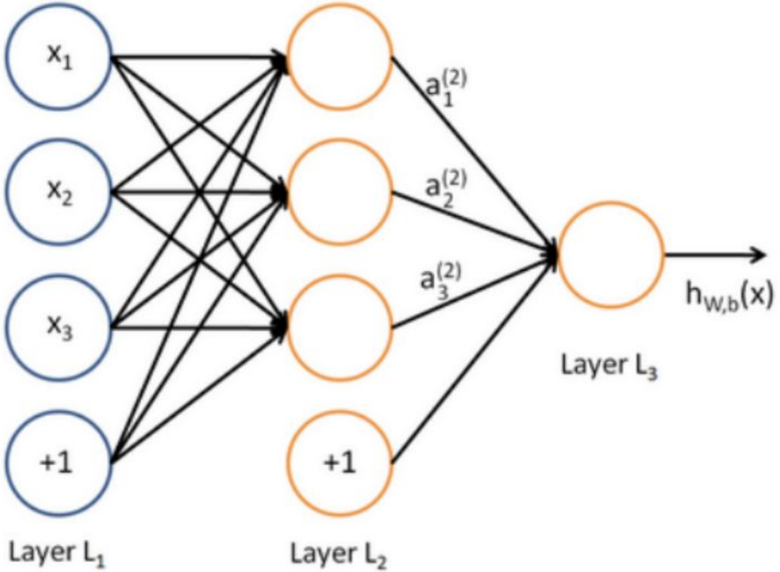
Neural Network

Detour: Neural Networks and Deep Learning

No need for feature engineering, the network learns higher or lower dimensions on itself



Logistic Regression



Neural Network

Detour: Neural Network

What do we mean by feature engineering?

Say for sentiment analysis,

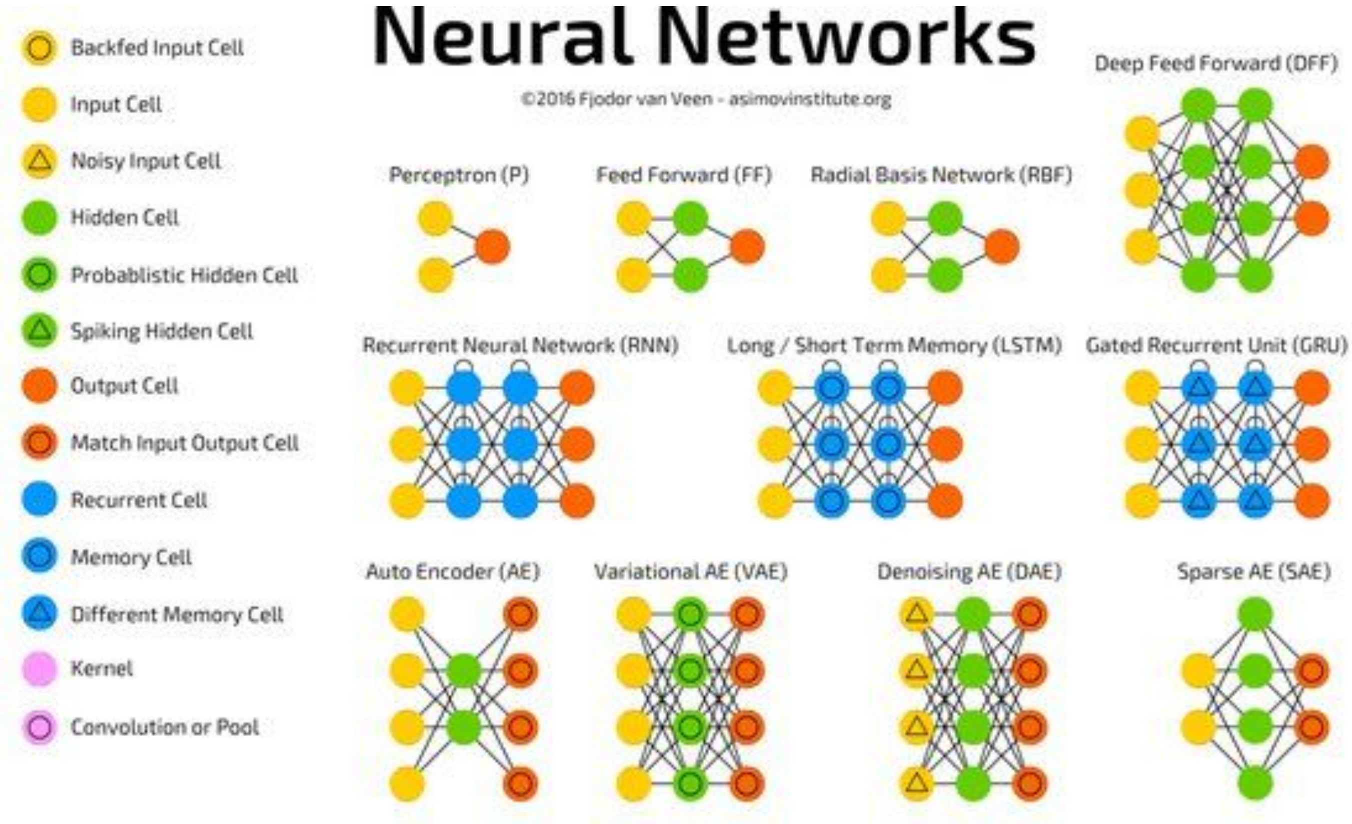
Table II. Summary of the Articles Employed a Supervised Method to Address TSA

Study	Task	Algorithms	Features	Dataset
Go et al. [2009]	TSA	NB, MaxEnt, SVM	unigrams, bigrams, POS	STS
Pak and Paroubek [2010]	TSA	MNB, SVM, CRF	unigrams, bigrams, trigrams, POS	own
Barbosa and Feng [2010]	TSA	SVM	meta-features (POS, polarity-MPQA), tweet syntax (i.e., retweet, hashtags, emoticons, links etc.)	own
Davidov et al. [2010]	TSA	kNN	word and n-gram based, punctuation-based, pattern-based	OC
Bakliwal et al. [2012]	TSA	SVM, NB	words' polarity, unigrams, bigrams, emoticons, hashtags, URLs, targets etc.	STS, Mejaj [Bora 2012]
Mohammad et al. [2013]	TSA	SVM	word/character n-grams, POS, caps, lexicons, punctuation, negation, tweet-based	SemEval-2013
Kiritchenko et al. [2014]	TSA	linear kernel SVM, MaxEnt	word/character n-grams, POS, caps, punctuation, emoticons, automatic sentiment lexicons, polarity, emphatic lengthening	SemEval-2013
Asiaee et al. [2012]	TSA	dictionary learning, WSVM, NB, kNN,		DETC
Agarwal et al. [2011]	TSA	SVM	POS, unigrams, DAL lexicon, caps, exclamation etc.	own
Aisopos et al. [2011]	TSA	MNB, C4.5 tree	n-grams	own
Kouloumpis et al. [2011]	TSA	AdaBoost	N-gram with lexicon features, twitter-based, POS	STS, ETC
Saif et al. [2012b]	TSA	NB	unigrams, POS, sentiment-topic, semantic features	STS, HCR, OMD
Hamdan et al. [2013]	TSA	SVM, NB	unigrams, concepts (DBPedia), verb groups/adjectives (WordNet) and senti-features (SentiWordNet)	SemEval-2013
Jiang et al. [2011]	entity-TSA	SVM	unigrams, emoticons, hashtags, punctuation the General Inquirer lexicon	own
Aston et al. [2014]	TSA	Perceptron with Best Learning	character n-grams	Sanders

Like It or Not: A Survey of Twitter Sentiment Analysis Methods, Anastasia Giachanou And Fabio Crestani, 2016

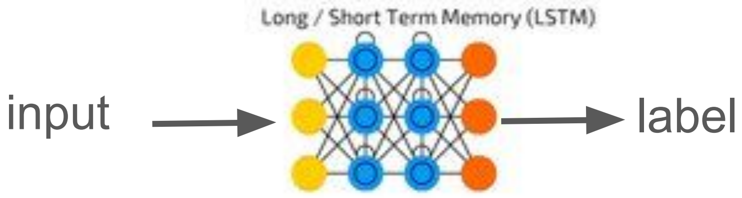
Method

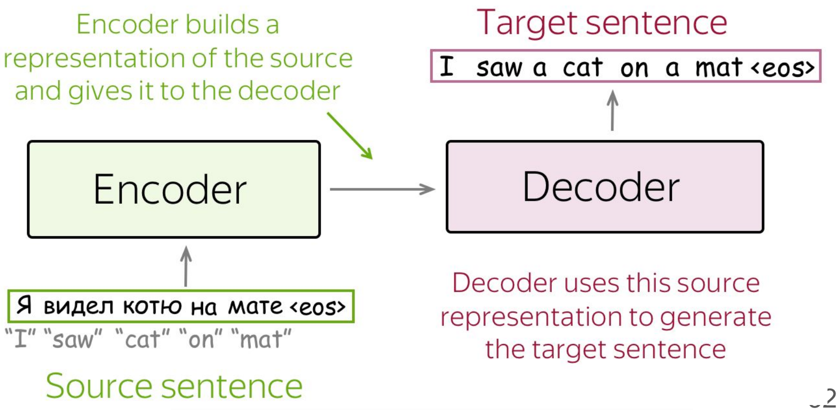
Detour: Neural Networks and Deep Learning



Deep Learning in NLP (till mid 2010's)

- Precludes need for explicit feature engineering
- Automatically creates complex representations
- Popular Classifiers are RNNs (GRUs and LSTMs)
- Classification is one strand of problems
- Another: sequence-to-sequence output, e.g., Machine Translation





- Lena Voita, NLP Course for you

Contextualized Embeddings (ELMo, CoVe)

Deep contextualized word representations

Matthew E. Peters†, Mark Neumann†, Mohit Iyyer†, Matt Gardner†,
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark*, Kenton Lee*, Luke Zettlemoyer†*

ington

<u>The idea behind Contextualized Embeddings</u>: Word *token* vectors instead of word *type*, with neural networks modeling left and right contexts

ge text cor-
LMo (Em-
esentations.
g contextu-
7; McCann

vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP

are deep, in the sense that they are a function of all of the internal layers of the biLM. More specifically, we learn a linear combination of the vectors stacked above each input word for each end task, which markedly improves performance over just using the top LSTM layer.

Contextualized Embeddings (ELMo, CoVe)

Deep contextualized word representations

Matthew E. Peters†, Mark Neumann†, Mohit Iyyer†, Matt Gardner†,
{matthewp,markn,mohiti,mattg}@allenai.org

Christopher Clark*, Kenton Lee*, Luke Zettlemoyer†*

ind Contextualized Embeddings: Word *token* vectors instead of word *type*, with neural networks modeling left and right contexts

ington

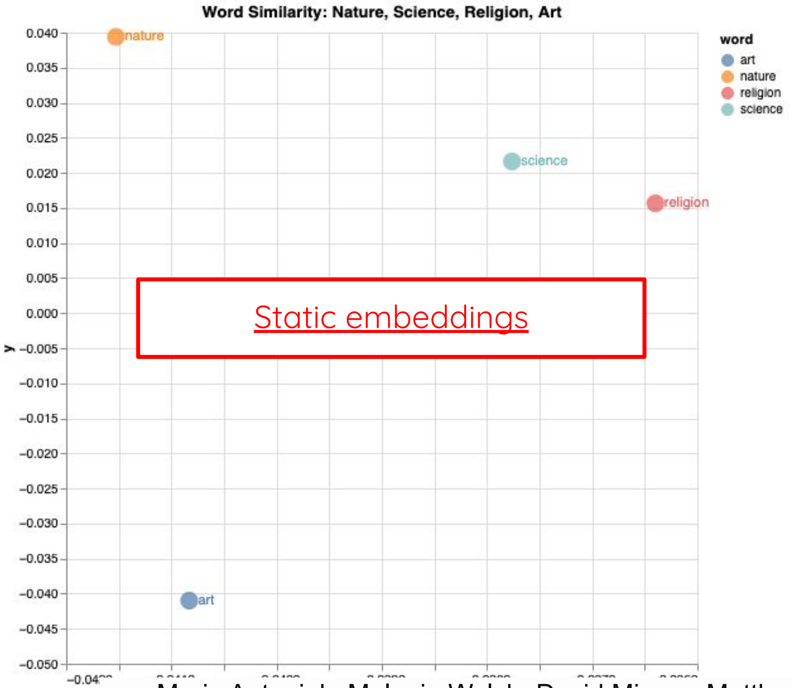
ge text cor-
LMo (Em-
esentations.
g contextu-
7; McCann
are deep, in

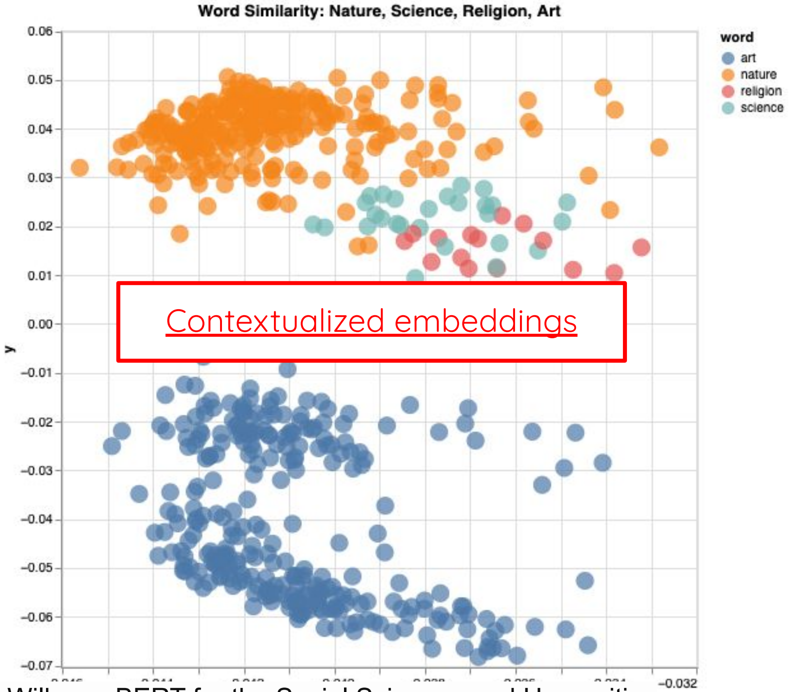
vary across linguistic contexts (i.e., to model polysemy). Our word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. We show that these representations can be easily added to existing models and significantly improve the state of the art across six challenging NLP

the sense that they are a function of all of the internal layers of the biLM. More specifically, we learn a linear combination of the vectors stacked above each input word for each end task, which markedly improves performance over just using the top LSTM layer.

Contextualized Embeddings (ELMo, CoVe)

Result: multiple vectors of a word depending on its context

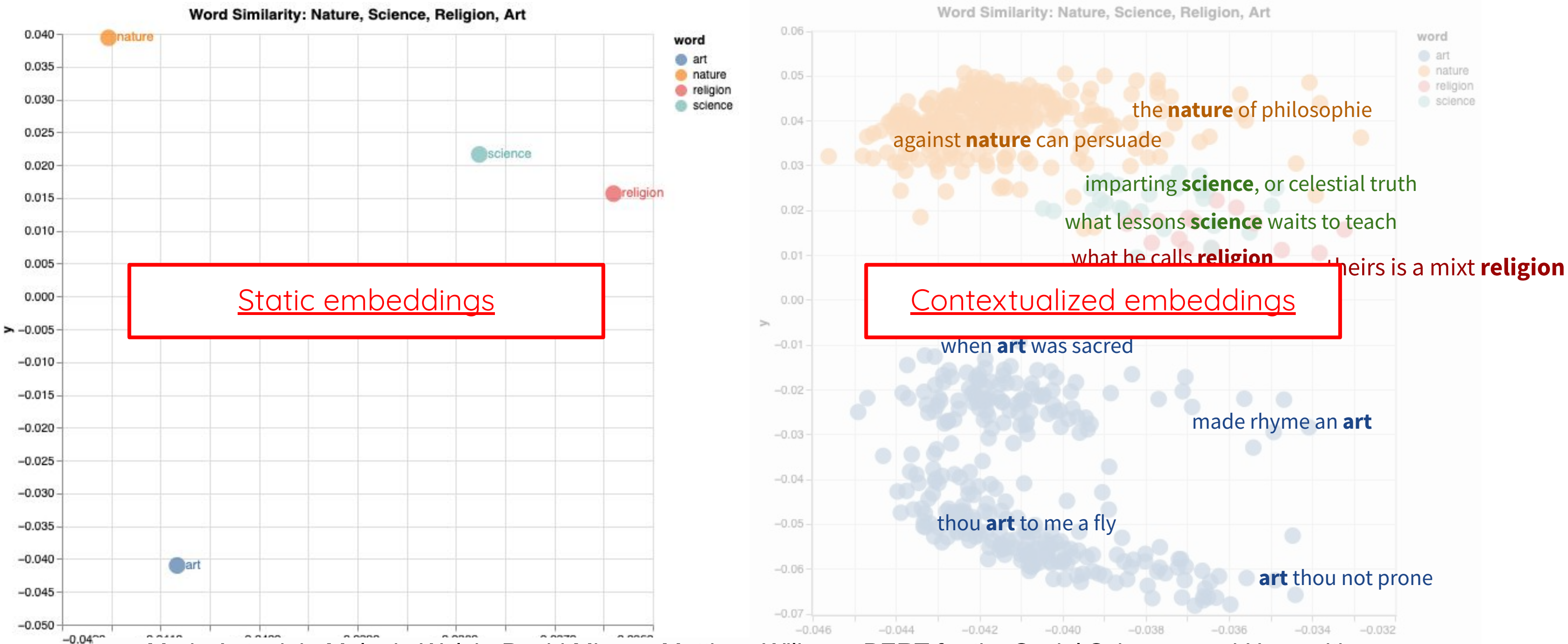




- Maria Antoniak, Melanie Walsh, David Mimno, Matthew Wilkens, BERT for the Social Sciences and Humanities

Contextualized Embeddings (ELMo, CoVe)

Result: multiple vectors of a word depending on its context



- Maria Antoniak, Melanie Walsh, David Mimno, Matthew Wilkens, BERT for the Social Sciences and Humanities

Transformers

- RNNs in architectures have limited context
- Fix: replace all components with attention

	Seq2seq without attention	Seq2seq with attention	Transformer
processing within encoder	RNN/CNN	RNN/CNN	attention
processing within decoder	RNN/CNN	RNN/CNN	attention
decoder-encoder interaction	static fixed-sized vector	attention	attention

- Lena Voita, NLP Course for you

BERT, GPT (the first one), ...

	GloVe, Word2Vec →	CoVe, ELMo →	GPT, BERT
What is encoded	words →	words in context	words in context
How it is used for downstream tasks	Input for task-specific models	Input for task-specific models →	Instead of task-specific models

Great idea 1

Great idea 2

- Lena Voita, NLP Course for you

BERT, GPT (the first one), ...

What is encoded

How it is used for downstream tasks

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the

The old vs. ‘new’ paradigm

OLD: Train a model on a task where we have labeled data

NEW:

1. **Pre-train** a model on a task where we have lots of data
   - *e.g., we have lots and lots of internet data*

Transfer learning!

2. **Fine-tune** the model on your downstream task
   - *e.g., a specific, curated dataset that you’re studying*

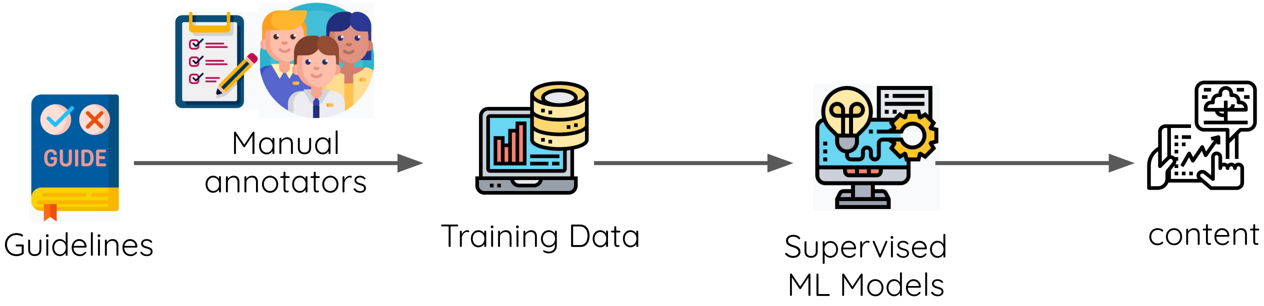
What is fine-tuning?

- Inputs:
  - pretrained model (usually a transformer, BERT and variants are still very popular)
  - Labeled dataset (rule-of-thumb, at least 1K examples)
- Adds a new layer containing classification parameters.
- All model parameters are updated to maximize the log probability of the labels.
- Output: classifier for your specific task

Let’s try it ourselves

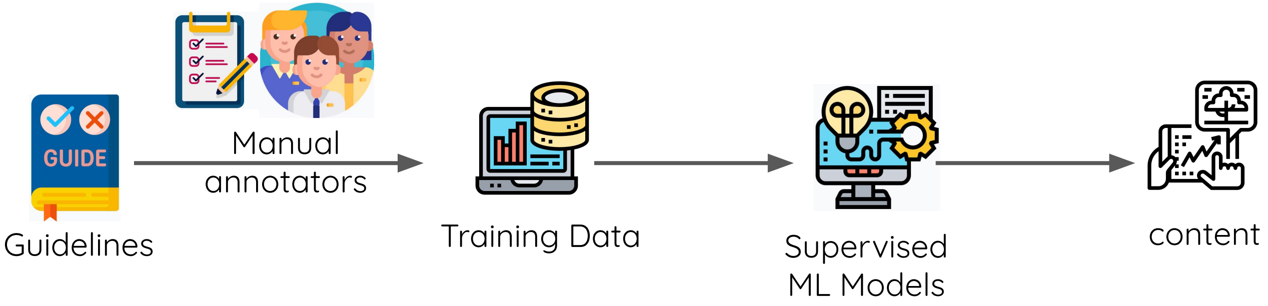
- unsupervised

- supervised



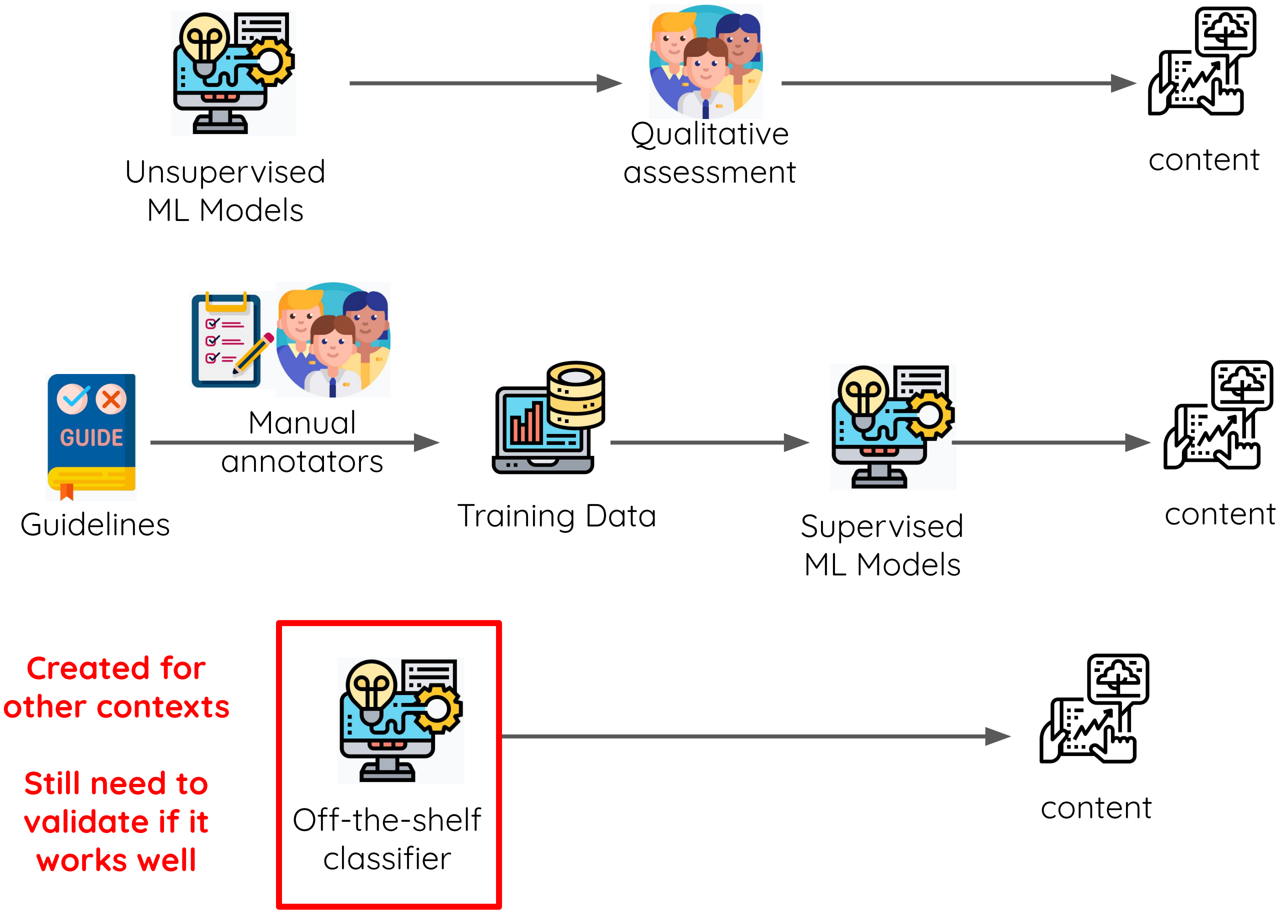
- unsupervised

- supervised

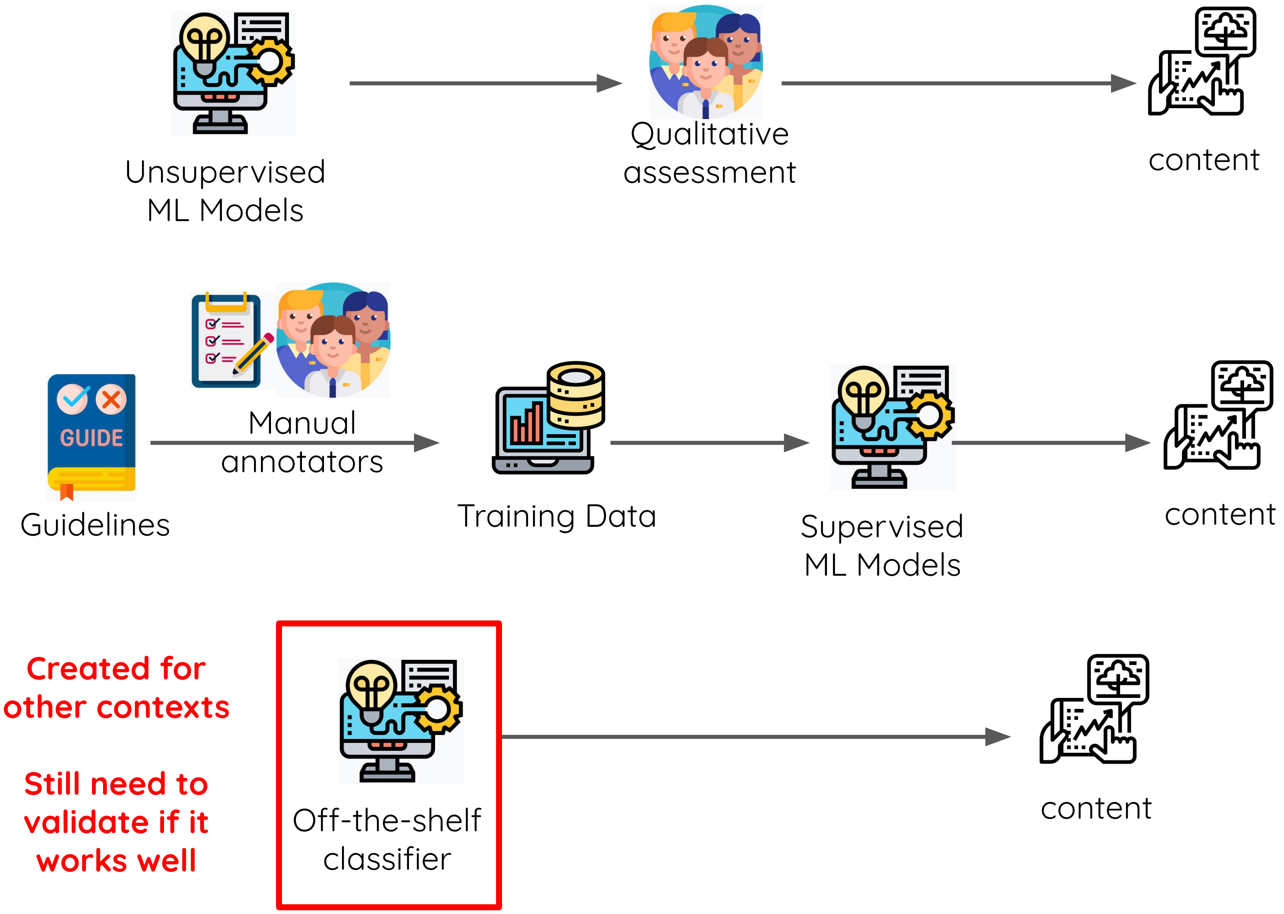


- ‘off-the-shelf’

- unsupervised
- supervised
- ‘off-the-shelf’



- unsupervised
- supervised
- ‘off-the-shelf’ e.g., MNLI, VADER for sentiment



Zero-Shot Natural Language Inference (NLI)

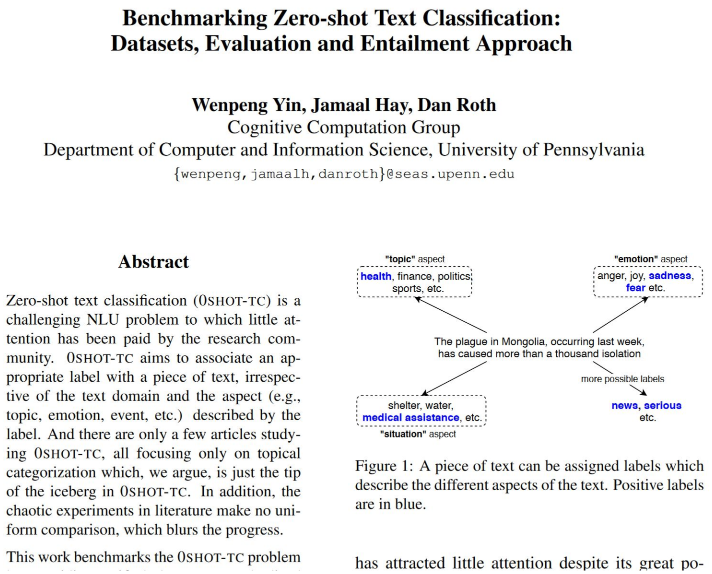
- The NLP task of detecting if a given premise justifies the given hypothesis
- Also called textual entailment
- Typically there are three classes:
  - Entailment
  - Contradiction
  - Neutral or none

Premise	Relation	Hypothesis
A turtle danced.	entails	A turtle moved.
turtle	contradicts	linguist
Every reptile danced.	neutral	A turtle ate.
Some turtles walk.	contradicts	No turtles move.
James Byron Dean refused to move without blue jeans.	entails	James Dean didn't dance without pants.
Mitsubishi Motors Corp's new vehicle sales in the US fell 46 percent in June.	contradicts	Mitsubishi's sales rose 46 percent.
Acme Corporation reported that its CEO resigned.	entails	Acme's CEO resigned.

https://web.stanford.edu/class/cs224u/2021/slides/cs224u-2021-nli-part1-handout.pdf

Zero-Shot Natural Language Inference (NLI)

- NLI is helpful for many other tasks:
  - Question answering
  - Summarization
  - …
- And now more recently, for zero-shot text classification
- Premise: Joe Biden's inauguration…
- Hypothesis: This example is about **politics**.



Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach

Wenpeng Yin, Jamaal Hay, Dan Roth
Cognitive Computation Group
Department of Computer and Information Science, University of Pennsylvania
{wenpeng,jamaalh,danroth}@seas.upenn.edu

Abstract

Zero-shot text classification (0SHOT-TC) is a challenging NLU problem to which little attention has been paid by the research community. 0SHOT-TC aims to associate an appropriate label with a piece of text, irrespective of the text domain and the aspect (e.g., topic, emotion, event, etc.) described by the label. And there are only a few articles studying 0SHOT-TC, all focusing only on topical categorization which, we argue, is just the tip of the iceberg in 0SHOT-TC. In addition, the chaotic experiments in literature make no uniform comparison, which blurs the progress.

This work benchmarks the 0SHOT-TC problem

Figure 1: A piece of text can be assigned labels which describe the different aspects of the text. Positive labels are in blue.

has attracted little attention despite its great po-

[Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach]

Let’s try it ourselves

Sentiment Analysis

Ribeiro et al. *EPJ Data Science* (2016) 5:23
DOI 10.1140/epjds/s13688-016-0085-1

EPJ.org

EPJ Data Science
a SpringerOpen Journal

REGULAR ARTICLE Open Access

CrossMark

SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods

Filipe N Ribeiro[1,2*], Matheus Araújo[1], Pollyanna Gonçalves[1], Marcos André Gonçalves[1] and Fabrício Benevenuto[1]

[*]Correspondence: filiperibeiro@dcc.ufmg.br
[1]Computer Science Department, Federal University of Minas Gerais, Belo Horizonte, Brazil
[2]Computer and Systems Department, Federal University of Ouro Preto, Joao Monlevade, Brazil

Abstract

In the last few years thousands of scientific papers have investigated sentiment analysis, several startups that measure opinions on real data have emerged and a number of innovative products related to this theme have been developed. There are multiple methods for measuring sentiments, including lexical-based and supervised machine learning methods. Despite the vast interest on the theme and wide popularity of some methods, it is unclear which one is better for identifying the polarity (i.e., positive or negative) of a message. Accordingly, there is a strong need to conduct a thorough apple-to-apple comparison of sentiment analysis methods, *as they are used in practice*, across multiple datasets originated from different data

Sentiment Analysis

Table 1 Overview of the sentence-level methods available in the literature

Name	Description	L	ML
Emoticons [20]	Messages containing positive/negative emoticons are positive/negative. Messages without emoticons are not classified.	✓	
Opinion Lexicon [2]	Focus on Product Reviews. Builds a Lexicon to predict polarity of product features phrases that are summarized to provide an overall score to that product feature.	✓	
Opinion Finder (MPQA) [22, 23]	Performs subjectivity analysis trough a framework with lexical analysis former and a machine learning approach latter.	✓	✓
SentiWordNet [24, 25]	Construction of a lexical resource for Opinion Mining based on WordNet [26]. The authors grouped adjectives, nouns, etc. in synonym sets (synsets) and associated three polarity scores (positive, negative and neutral) for each one.	✓	✓
LIWC [7]	An acronym for Linguistic Inquiry and Word Count, LIWC is a text analysis paid tool to evaluate emotional, cognitive, and structural components of a given text. It uses a dictionary with words classified into categories (anxiety, health, leisure, etc.). An updated version was launched in 2015.	✓	
Sentiment140 [27]	Sentiment140 (previously known as 'Twitter Sentiment') was proposed as an ensemble of three classifiers (Naive Bayes, Maximum Entropy, and SVM) built with a huge amount of tweets containing emoticons collected by the authors. It has been improved and transformed into a [illegible]		
SenticNet [28]	[illegible] sense [illegible] om text it	✓	
AFINN [29] - a ne[illegible] ANEW	[illegible] ings and obscene words. AFINN can be considered as an expansion of ANEW [30], a dictionary created to provides emotional ratings for English words. ANEW dictionary rates words in terms of pleasure, arousal and dominance.	✓	
SO-CAL [31]	Creates a new Lexicon with unigrams (verbs, adverbs, nouns and adjectives) and multi-grams (phrasal verbs and intensifiers) hand ranked with scale +5 (strongly positive) to −5 (strongly negative). Authors also included part of speech processing, negation and intensifiers.	✓	
Emoticons DS (Distant Supervision) [32]	Creates a scored lexicon based on a large dataset of tweets. Its based on the frequency each lexicon occurs with positive or negative emotions.	✓	
NRC Hashtag [33]	Builds a lexicon dictionary using a Distant Supervised Approach. In a nutshell it uses known hashtags (i.e. #joy, #happy, etc.) to 'classify' the tweet. Afterwards, it verifies frequency each specific n-gram occurs in a emotion and calculates its Strong of Association with that emotion.	✓	
Pattern.en [34]	Python Programming Package (toolkit) to deal with NLP, Web Mining and Sentiment Analysis. Sentiment analysis is provided through averaging scores from adjectives in the sentence according to a bundle lexicon of adjective.	✓	
SASA [35]	Detects public sentiments on Twitter during the 2012 U.S. presidential election. It is based on the statistical model obtained from the classifier Naive Bayes on unigram features. It also explores emoticons and exclamations.		✓
PANAS-t [8]	Detects mood fluctuations of users on Twitter. The method consists of an adapted version (PANAS) Positive Affect Negative Affect Scale [36], well-known method in psychology with a large set of words, each of them associated with one from eleven moods such as surprise, fear, guilt, etc.	✓	
Emolex [37]	Builds a general sentiment Lexicon crowdsourcing supported. Each entry lists the association of a token with 8 basic sentiments: joy, sadness, anger, etc. defined by [38]. Proposed Lexicon includes unigrams and bigrams from Macquarie Thesaurus and also words from GI and WordNet.	✓	
USent [39]	Infer additional reviews user ratings by performing sentiment analysis (SA) of user comments and integrating its output in a nearest neighbor (NN) model that provides multimedia recommendations over TED talks.	✓	✓

So many options!

Data Analysis: Sentiment Analysis

- VADER: Valence Aware Dictionary for sEntiment Reasoning
- Gold-standard sentiment dictionary + preprocessing engine
- positive score, negative score, neutral score and *complex* score where one can fix a threshold
- usually: negative < -0.1, 0.1 < positive

> **VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text**
>
> **C.J. Hutto** **Eric Gilbert**
>
> Georgia Institute of Technology, Atlanta, GA 30032
>
> cjhutto@gatech.edu gilbert@cc.gatech.edu

Let's try it ourselves