Hate Speech Classification Using BERT

LING 575: Group 2 “The Cool Kids”
Bill Presant, Courtney Mansfield, David Nielsen, Preeti Mohan, Simola Nayak, Ben Longwill

Our Team

Courtney Mansfield

David Nielsen

Preeti Mohan

Ben Longwill

Simola Nayak

Bill Presant

Outline

1. Introduction
2. The task - What is hate speech?
3. The model - How has BERT been used in this task before?
4. The analysis - Methods for transfer learning and analyzing neurons

Our Project

Can BERT predict hate speech across multiple languages? How does BERT encode information about hate speech in its layers/neurons?

Methods

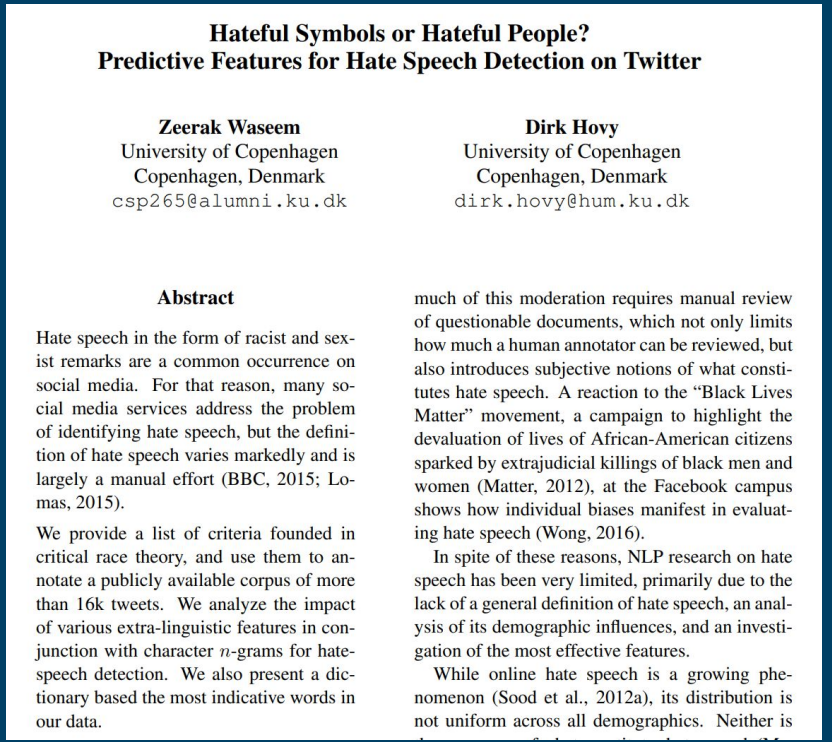
- Diagnostic classifier
- Visualization of individual neurons

Applications

- Understanding how hate speech is classified might help to reduce bias
- Applying multilingual models for hate speech classification in low resource languages

What is hate speech?

And how do we detect it?



Hateful Symbols or Hateful People?
Predictive Features for Hate Speech Detection on Twitter

Zeerak Waseem
University of Copenhagen
Copenhagen, Denmark
csp265@alumni.ku.dk

Dirk Hovy
University of Copenhagen
Copenhagen, Denmark
dirk.hovy@hum.ku.dk

Abstract

Hate speech in the form of racist and sexist remarks are a common occurrence on social media. For that reason, many social media services address the problem of identifying hate speech, but the definition of hate speech varies markedly and is largely a manual effort (BBC, 2015; Lomas, 2015).

We provide a list of criteria founded in critical race theory, and use them to annotate a publicly available corpus of more than 16k tweets. We analyze the impact of various extra-linguistic features in conjunction with character n-grams for hate-speech detection. We also present a dictionary based the most indicative words in our data.

much of this moderation requires manual review of questionable documents, which not only limits how much a human annotator can be reviewed, but also introduces subjective notions of what constitutes hate speech. A reaction to the "Black Lives Matter" movement, a campaign to highlight the devaluation of lives of African-American citizens sparked by extrajudicial killings of black men and women (Matter, 2012), at the Facebook campus shows how individual biases manifest in evaluating hate speech (Wong, 2016).

In spite of these reasons, NLP research on hate speech has been very limited, primarily due to the lack of a general definition of hate speech, an analysis of its demographic influences, and an investigation of the most effective features.

While online hate speech is a growing phenomenon (Sood et al., 2012a), its distribution is not uniform across all demographics. Neither is

Waseem & Hovy (2016)

- List of hate speech criteria based on critical race theory
- Provides annotated dataset of 16k tweets
  - Annotated into categories of *sexist*, *racist*, or *neither* (hate speech that does not fall in either of the previous categories)
- Provide dictionary of most indicative words
- Discusses impact of extra-linguistic features in conjunction with n-grams from tweets that contain hate speech

Does hate speech classification matter?

"Hate speech is defined as any communication that disparages a person or a group on the basis of some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic." (Nockleby, 2000)

- Common on the internet
- Exists strong connection between hate speech and hate crimes
- Could manifest into severe threats to individuals
- Early detection = prevention programs

Waseem & Hovy (2016) continued

Why has research been limited?

- Hate speech is complex
  - Racial/sextist slurs are easy to identify
  - Does hate speech always contain slurs or bad language?
- Human identification/annotation is complicated
  - Defining hate speech is difficult
  - Not uniform across all demographics -- different levels of knowledge/exposure
  - People's opinions are biased
  - Similar to identifying privilege, requires critical thinking process and clear decision list
  - Unclear how to handle inter-annotator disagreement
  - Stress for human annotators

    https://www.theverge.com/2019/12/16/21021005/google-youtube-moderators-ptsd-accenture-violent-disturbing-content-interviews-video

Everyone else, despite our commentary, has fought hard too. It's not just you, Kat. #mkr

Waseem & Hovy (2016) continued

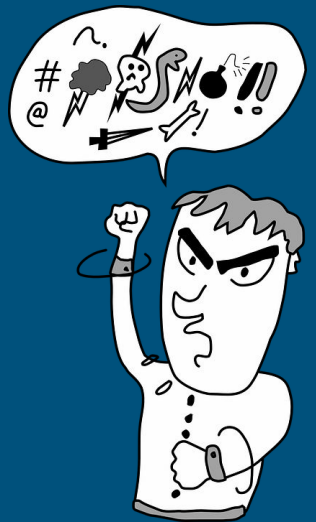
A tweet is offensive if it...

Waseem & Hovy (2016) continued

1) Uses a sexist or racial slur
2) Attacks a minority
3) Seeks to silence a minority
4) Criticizes a minority
    a) Without a well founded argument
5) Promotes (but does not directly use) hate speech or violent crime
6) Criticizes a minority and uses a straw man argument
7) Blatantly misrepresents truth or seeks to distort views on a minority with unfounded claims
8) Shows support of problematic hashtags.
9) Negatively stereotypes a minority
10) Defends xenophobia or sexism
11) Contains a screen name that is offensive, as per the previous criteria, the tweet is ambiguous (at best), and the tweet is on a topic that satisfies any of the above criteria.

Methods for classifying hate speech

- Manual annotation
- Automatic classification
  - Lists of keywords
  - SVM, naive bayes, RNN, CNN, LSTM...
  - Recently, BERT/pre-trained models



In NLP, binary classification, but can include type or degree of aggression

Waseem & Hovy (2016) continued

Demographics

- **Gender**
  - Names in user profile text, real name, or username compared to list of known male/female names
  - Also pronouns, honorifics, and gender specific nouns
  - 47.64% of users could not be identified
- Table 1 results heavily skewed towards men
- Congruent with Roberts et al. (2013) and Watch, (2014)
  - Hate speech is precursor to hate crime
  - 75%/87% of perps in Caribbean/Asian hate crime were men
- **Geography**
  - Only 2% of users disclose location
  - Determined from tweet timezone metadata or username/name
  - If time zone identified, long/lat coords added to feature set
  - If nominal location used, also added as feature

	All	Racism	Sexism	Neither
Men	50.08%	33.33%	50.24%	50.92%
Women	02.26%	0.00 %	02.28%	01.74%
Unidentified	47.64%	66.66%	47.47%	47.32%

Table 1: Distribution of genders in hate-speech documents.

Waseem & Hovy (2016) continued

Corpus / Dataset

- Data is collected via bootstrapping from Twitter data:
  - Tweets containing **common slurs**
  - Tweets containing **terms + hashtags** commonly used in hate speech
- Model should learn to **recognize hate speech** rather than **recognizing individual slurs/terms**
  - Dataset should include negative samples that contain these slurs/terms/hashtags
  - Avoids false positives:
    - Critical discussion of racism + sexism
    - Reclaimed slurs
    - Sarcasm
- Data is **manually annotated** by annotators hired by authors
  - Not an option for projects with limited time/funds

Lexical Distribution

- Removes stop words and special chars with the exception of “not”
  - Including RT, screen names, and punctuation
- Includes avg/total tweet lengths & length of user descriptions as features
- Includes inferred location as a feature (timezone/timestamp; mentions of locations in tweet)
  - Actual tagged locations are also used, but uncommon (<5% of tweets)
- Constructs 10 most frequently occuring words per class
- Words in separate classes differ greatly
- Table 2 shows sampling effect
  - Tweets tagged as sexism are mostly by viewers of ‘My Kitchen Rules’ (an AUS. t.v. show)
  - Tweets tagged as racism often pertain to topics of Islam and Judaism

	Racism	Sexism	None
Mean	60.47	52.93	47.95
Std.	17.44	21.16	23.43
Min.	11.00	2.00	2.00
Max.	115.00	118.00	129.00

Table 3: Overview of lengths in characters, subtracting spaces.

Sexism	Distribution	Racism	Distribution
not	1.83%	islam	1.44%
sexist	1.68%	muslims	1.01%
#mkr	1.57%	muslim	0.65%
women	0.83%	not	0.53%
kat	0.57%	mohammed	0.52%
girls	0.48%	religion	0.40%
like	0.42%	isis	0.38%
call	0.36%	jews	0.37%
#notsexist	0.36%	prophet	0.36%
female	0.34%	#islam	0.35%

Table 2: Distribution of ten most frequently occurring terms

Waseem & Hovy (2016) continued

The problem with words

- tr0lls c4n g3t 4r0und f1lt3rs
- Sparsity of words
- Emergent words

Character n-grams with lengths up to 4 and gender (of tweet author) were the features that were found to provide the best results.

Feature (sexism)	Feature (racism)
'xist'	'sl'
'sexi'	'sla'
'ka'	'slam'
'sex'	'isla'
'kat'	'l'
'exis'	'a'
'xis'	'isl'
'exi'	'lam'
'xi'	'i'
'bitc'	'e'
'ist'	'mu'
'bit'	's'
'itch'	'am'
'itc'	'm'
'fem'	'la'
'ex'	'is'
'bi'	'slim'
'irl'	'musl'
'wom'	'usli'
'girl'	'lim'

Table 5: Most indicative character n-gram features for hate-speech detection

Waseem & Hovy (2016) continued

Effect of Features on Model Performance

- Logistic regression and 10-fold cross validation
- Gender gives improvement; however, it is not statistically significant
- Gender and location is significant ($p < 0.05$)
- Demographic results attributed to lack of coverage

	char n-grams	+gender	+gender +loc	word n-grams
F1	73.89	73.93	73.62*	64.58
Precision	72.87%	72.93%	72.58%	64.39%
Recall	77.75%	77.74%	77.43%	71.93%

Waseem & Hovy (2016) continued

Conclusions

- Character n-grams have clear advantages in hate speech detection
  - Different sets of n-grams are useful for classifying ‘sexism’ vs ‘racism’
- While demographic data could prove useful, lack of coverage is an issue
- Hate speech detection is a complex issue and requires more research

How has BERT been used in this task before?

MC-BERT4HATE: Hate Speech Detection using Multi-channel BERT for Different Languages and Translations

Hajung Sohn
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju, Korea
hajungsohn@gist.ac.kr

Hyunju Lee
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju, Korea
hyunjulee@gist.ac.kr

***Abstract*—The growth of social networking services (SNS) has altered the way and scale of communication in cyberspace. However, the amount of online hate speech is increasing because of the anonymity and mobility such services provide. As manual hate speech detection by human annotators is both costly and time consuming, there are needs to develop an algorithm for automatic recognition. Transfering knowledge by fine-tuning a pre-trained language model has been shown to be effective for improving many downstream tasks in the field of natural language processing. The Bidirectional Encoder Representations from Transformers (BERT) is a language model that is pre-trained to learn deep bidirectional representations from a large corpus. In this paper, we propose a multi-channel model with three versions of BERT (MC-BERT), the English, Chinese, and multilingual BERTs for hate speech detection. We also explored the usage of translations as additional input by translating training and test sentences to the corresponding languages required for different BERT models. We used three datasets in non-English languages to compare our model with previous approaches including the 2019 SemEval HatEval Spanish dataset, 2018 GermEval shared task on the identification of Offensive Language dataset, and 2018 EvalIta HaSpeeDe Italian dataset. Finally, we were able to achieve the state-of-the-art or comparable performance on these datasets by conducting thorough experiments.**

***Index Terms*—BERT, Deep Learning, Hate speech, Sentence Classification, Social Networking Services, Transfer Learning**

I. Introduction

- *@USER How is she hiding her ugly personality. She is the worst.*
- *Conservatism101 It's not about our disagreements with Conservatives. Its that Conservatives can't debate honestly, and they have no integrity. Whatever gets them thru today, is all that matters to them. They're fundamentally dishonest people. URL*

This type of language is considered as a social problem because most of the contents aim disadvantaged social groups and can further lead to the development of organized hate-based activities. With the increase in the social impact of hate speech over the past years, the interests of the research commu¨ity, including governments, SNS companies, and individual researchers, in recognizing hate speech, have grown. Although many SNS companies have hired human annotators to manually filter out hate speech, they are still criticized for not doing enough [32]. As the manual detection of hate speech is both costly and time consuming, automatic detection methods are required.

There are some characteristics in hate speech that complicate its automatic identification without the involvement of human annotators. First, there is no absolute standard for what comprises a hate speech. The standard of offensive languages can vary based on country, time, culture, and political propen-

Sohn & Lee (2019) - Setup

- Transfer learning using a combination of 3 versions of BERT (MC-BERT)
  - English
  - Chinese
  - Multilingual
- Tested on 3 non-English datasets for hate speech detection
  - Spanish (HatEval)
  - German (GermEval)
  - Italian (HaSpeeDa)

Sohn & Lee (2019) - Transfer Learning

- Fine tuning task
  - Binary classification Hate/non-hate speech
- Simple classifier built on top of Multilingual BERT
- Using translated text to supplement source text
  - Translated tweets into English or Chinese
  - Built simple classifier with corresponding BERT
  - Also built multichannel model combines source with these translations

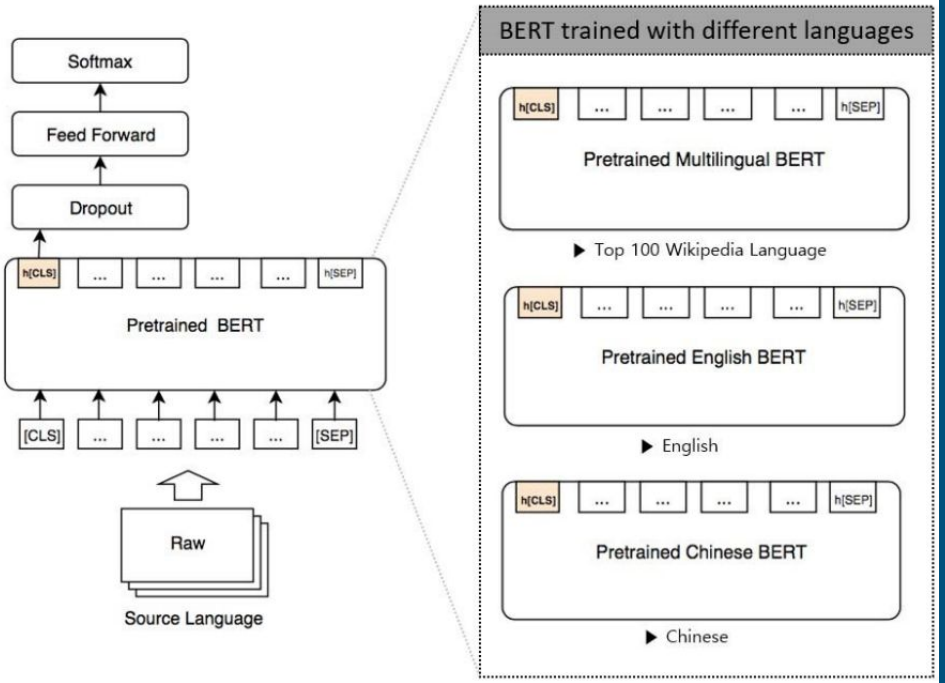


Fig. 4. Baseline model: Fine-tuning BERT for different languages

Sohn & Lee (2019) continued

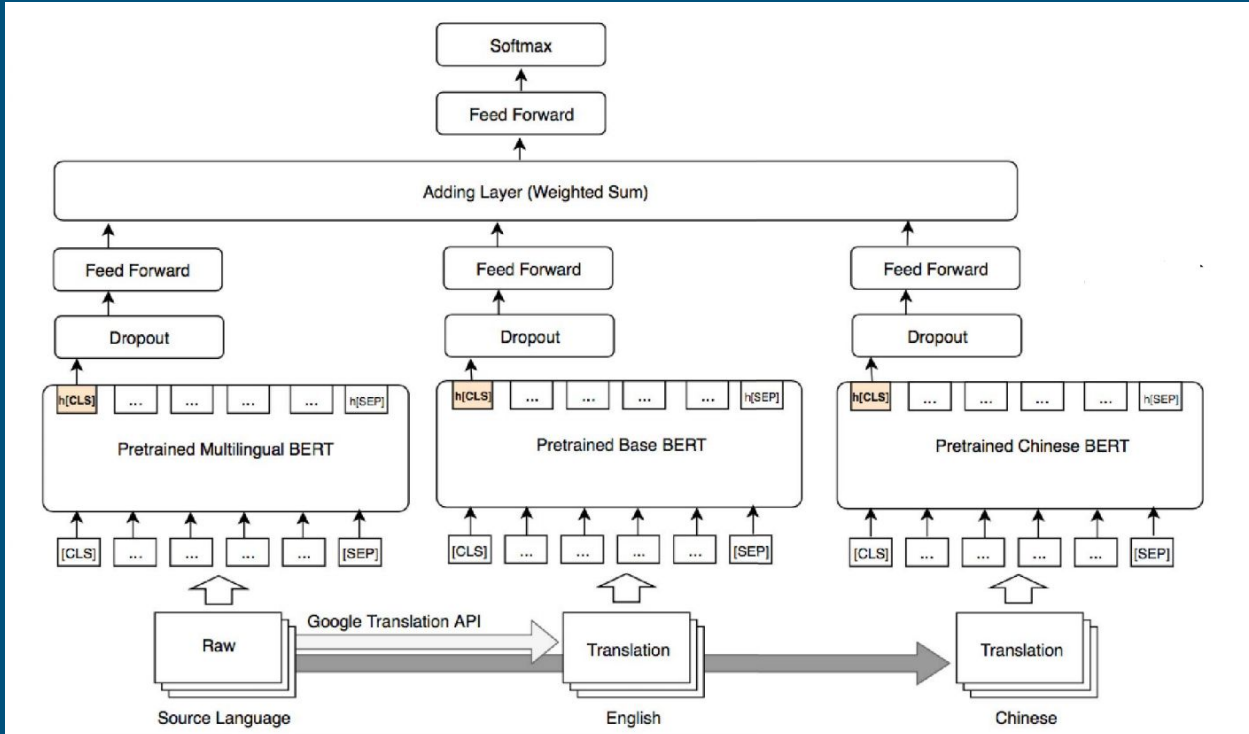


Fig. 5. Multi-channel BERT for different languages (MC-BERT)

Sohn & Lee (2019) continued

Sohn & Lee (2019) - Results

- Accuracy and macro-F1 score for
  - Published state of the art
  - English BERT
  - Chinese BERT
  - Multi-lingual BERT
  - Multichannel BERT

Spanish (HatEval)

HatEval Results

Method	Accuracy	F1 Macro
SVC Baseline [37]	0.705	0.701
Glove + LSTM	0.716	0.710
(SOTA) BoW+ BoC + fasttext + SVM (Perez and Luque, 2019) [18]	0.731	0.730
English BERT fine-tune	0.752	0.748
Multilingual BERT fine-tune	0.755	0.751
Chinese BERT fine-tune	0.700	0.690
Multi-channel BERT fine-tune	**0.769**	**0.766**

Sohn & Lee (2019) continued

German (GermEval)

GermEval Results

Method	Accuracy	F1 Macro
Fasttext + LSTM	0.700	0.638
(SOTA) ngram + ensemble of LR and RF (Montani and Schuller, 2018) [42]	0.795	0.767
English BERT fine-tune	0.798	**0.770**
Multilingual BERT fine-tune	0.771	0.732
Chinese BERT fine-tune	0.760	0.720
Multi-channel BERT fine-tune	**0.801**	0.764

Sohn & Lee (2019) continued

Italian (HaSpeeDe)

HaSpeeDe Results

Method	Accuracy	F1 Macro
Fasttext + LSTM	0.783	0.753
Fasttext + SVM	-	0.774
(SOTA) SVM + (bi)LSTM + additional data (Cimino et al., 2018) [43]	-	**0.799**
English BERT fine-tune	0.798	0.773
Multilingual BERT fine-tune	**0.822**	**0.799**
Chinese BERT fine-tune	0.799	0.775
Multi-channel BERT fine-tune	0.800	0.775

Sohn & Lee (2019) continued

Sohn & Lee (2019) - Results

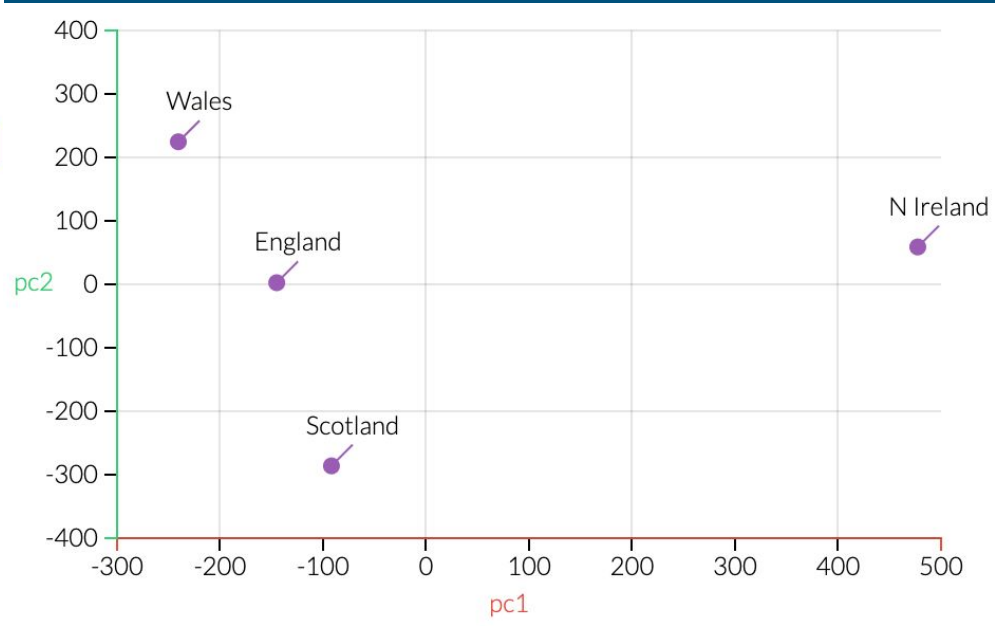
- BERT-based transfer learning works for hate speech classification!
- It not only works, but at least equals state of the art
- The translated text is helpful to the task regardless of errors in the translation
  - Except for Italian!
- But why does it work?

Sohn & Lee (2019) - Visualizing the task

- Principal Component Analysis was done on the input to the final feedforward layer for each model.
- This 2-dimensional representation shows how separable hate/non-hate data is before the final classification layers.

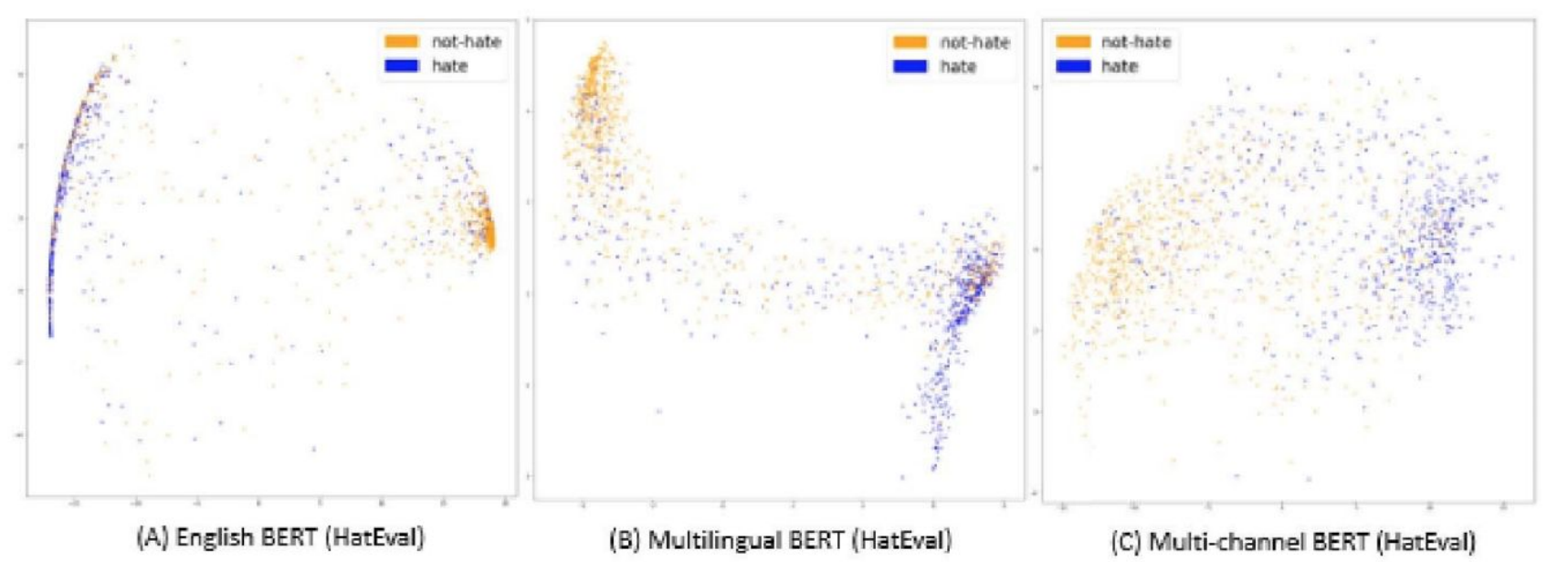
Example PCA - 17 dimensions to 2

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175



Example from
http://setosa.io/ev/principal-component-analysis/

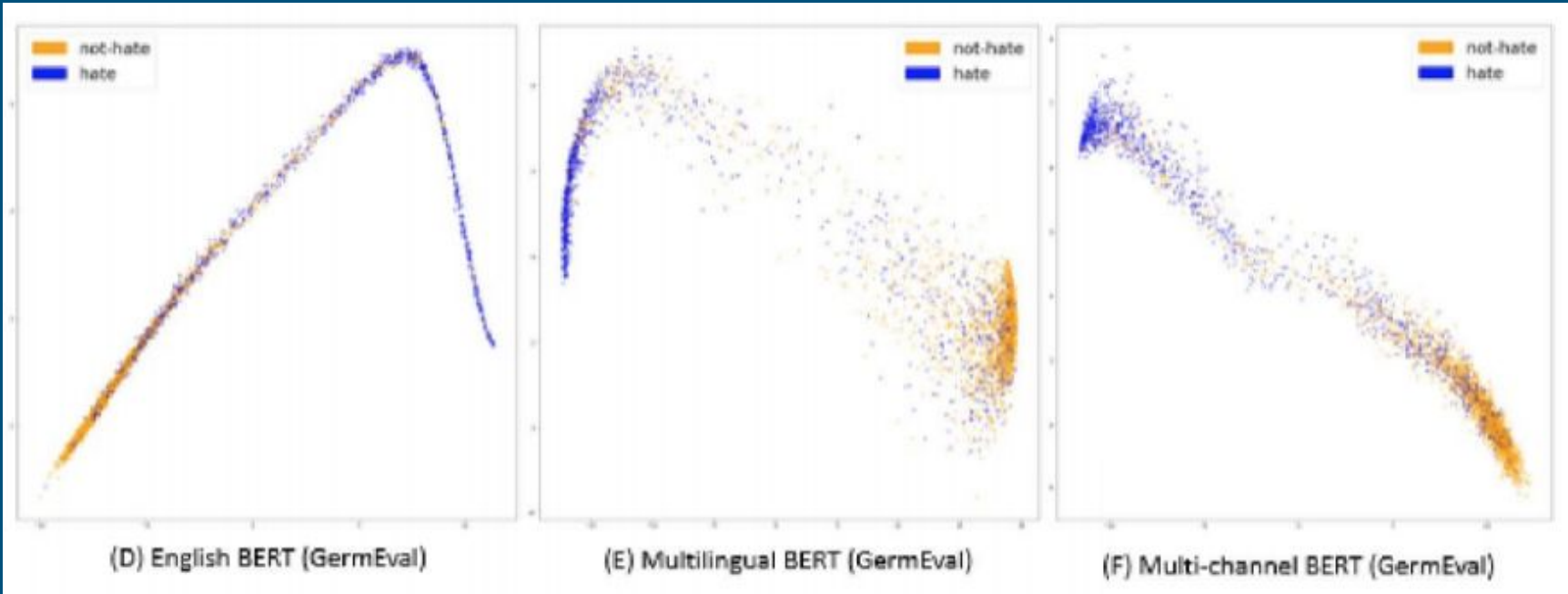
Spanish (HatEval)



(A) English BERT (HatEval) (B) Multilingual BERT (HatEval) (C) Multi-channel BERT (HatEval)

Accuracy	F1 Macro	Accuracy	F1 Macro	Accuracy	F1 Macro
0.752	0.748	0.755	0.751	**0.768**	**0.766**

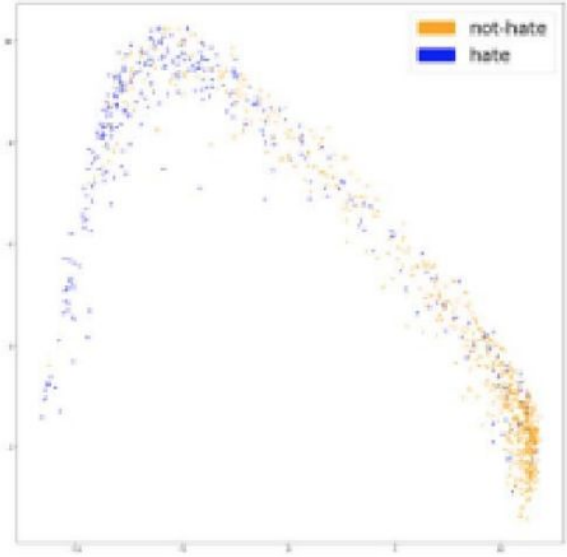
German (GermEval)



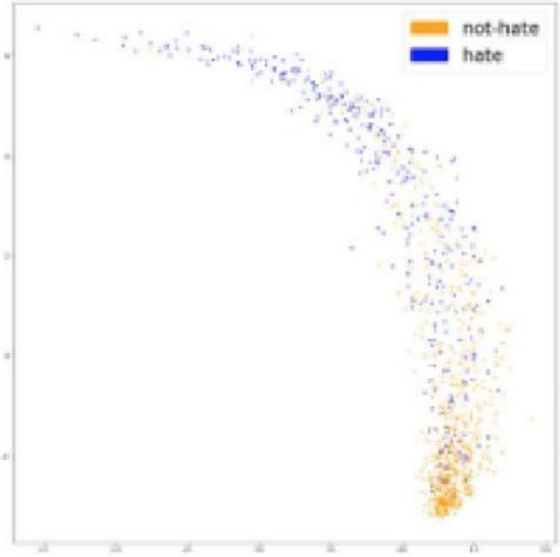
(D) English BERT (GermEval) (E) Multilingual BERT (GermEval) (F) Multi-channel BERT (GermEval)

Accuracy	F1 Macro	Accuracy	F1 Macro	Accuracy	F1 Macro
0.798	**0.770**	0.771	0.732	**0.801**	0.764

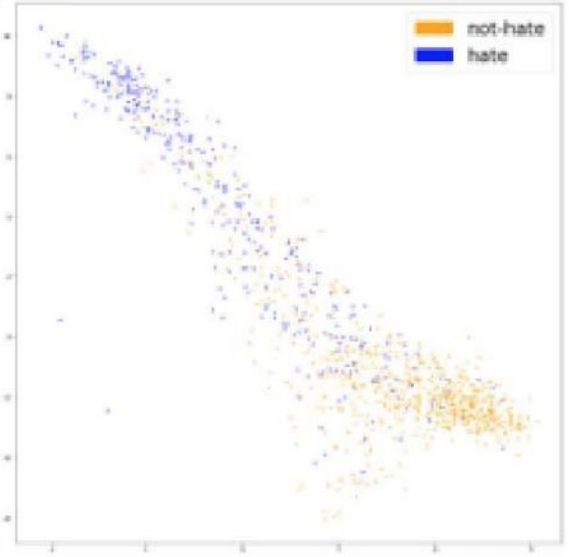
Italian (HaSpeeDe)



(G) English BERT (HaSpeeDe)



(H) Multilingual BERT (HaSpeeDe)



(I) Multi-channel BERT (HaSpeeDe)

Accuracy	F1 Macro	Accuracy	F1 Macro	Accuracy	F1 Macro
0.798	0.773	**0.822**	**0.799**	0.800	0.775

Sohn & Lee (2019) - Conclusion

- BERT-based transfer learning works for hate speech classification!
- The PCA shows some underlying structure is being found by these models
- Beyond PCA graphs, not much discussion of *why* the above results hold
  - Motivation for us to try and understand the whys

Methods for analyzing individual neurons

What Is One Grain of Sand in the Desert?
Analyzing Individual Neurons in Deep NLP Models

Fahim Dalvi,[*1] Nadir Durrani,[*1] Hassan Sajjad,[*1]
Yonatan Belinkov,[2] Anthony Bau,[2] James Glass[2]

[1]Qatar Computing Research Institute, HBKU Research Complex, Doha 5825, Qatar
[2]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
{faimaduddin,ndurrani,hsajjad}@qf.org.qa
{belinkov,abau,glass}@mit.edu

Abstract

Despite the remarkable evolution of deep neural networks in natural language processing (NLP), their interpretability remains a challenge. Previous work largely focused on what these models learn at the representation level. We break this analysis down further and study individual dimensions (neurons) in the vector representation learned by end-to-end neural models in NLP tasks. We propose two methods: *Linguistic Correlation Analysis*, based on a supervised method to extract the most relevant neurons with respect to an extrinsic task, and *Cross-model Correlation Analysis*, an unsupervised method to extract salient neurons w.r.t. the model itself. We evaluate the effectiveness of our techniques by ablating the identified neurons and reevaluating the network's performance for two tasks: neural machine translation (NMT) and neural language modeling (NLM). We further present a comprehensive analysis of neurons with the aim to address the following questions: i) how localized or distributed are different linguistic properties in the models? ii) are certain neurons exclusive to some properties and not others? iii) is the information more or less distributed in NMT vs. NLM? and iv) how important are the neurons identified through the linguistic correlation method to the overall task? Our code is publicly available[1] as part of the NeuroX toolkit (Dalvi et al. 2019).

Introduction

and predict a property of interest such as morphological features. This approach has also been applied for analyzing word and sentence embeddings (Qian, Qiu, and Huang 2016b; Adi et al. 2016), and hidden states in NMT models (Shi, Padhi, and Knight 2016; Belinkov et al. 2017a). The analyses reveal that neural vector representations often contain substantial amount of linguistic information. Most of this work, however, targets the whole vector representation, neglecting the individual dimensions in the embeddings. In contrast, much work in computer vision investigates properties encoded in individual neurons or filters (Zeiler and Fergus 2014; Zhou et al. 2016).

We address this gap by studying individual dimensions (neurons) in the vector representations learned by end-to-end neural models. We aim to increase model transparency by identifying specific dimensions that are responsible for particular properties. We thus strive for post-hoc decomposibility, in the sense of (Lipton 2016). That is, we analyze models after they have been trained, in order to uncover the importance of their individual parameters. This kind of analysis is important for improving understanding of the inner workings of neural networks. It also has potential applications in model distillation (e.g., by removing unimportant neurons), neural architecture search (by guiding the search with important neurons), and mitigating model bias (by identifying neurons responsible for sensitive attributes

Dalvi et al. (2018)

- Provides 2 methods for identifying meaningful neurons (1 supervised, 1 unsupervised)
- Are the ‘meaningful’ neurons important to each task?
- What linguistic properties are encoded in neurons?
  - Show how linguistic information is distributed across neurons
  - Visualize the output of specific neurons
- How does the task affect the saliency of the neurons? (LM vs. NMT)

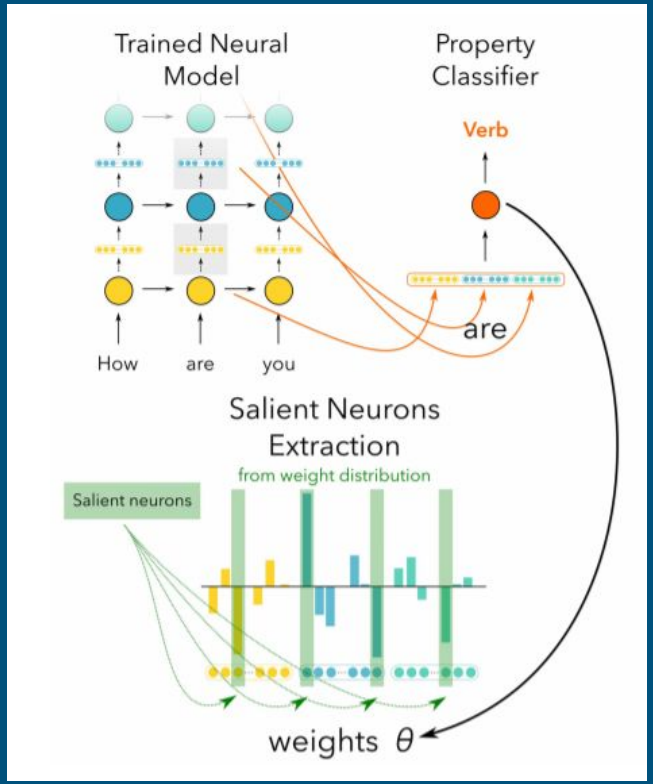
Two Methods:

1) *Linguistic Correlation Analysis (LCA)*
   a) Supervised classification
   b) Correlation analysis on linguistic properties deemed important. Extract individual neurons that capture these important properties.
2) *Cross-model Correlation Analysis (CCA)*
   a) Unsupervised classification
   b) Search for neurons that have similar patterns across independently trained networks to look for characteristics that may make a neuron important.

Dalvi et al. (2018) continued

Linguistic Correlation Analysis (LCA)

1. Extract latent representations from the Trained Neural Model (e.g. NMT encodings)
2. Representations are inputs to a logistic regression classifier (i.e. a transfer learning **probing task**)
3. Learned weights in the classifier are ranked to measure the importance of each neuron



Dalvi et al. (2018) continued

Choosing Salient Neurons

- Neurons are ranked according to their correlation with each particular category, and with increasing

- Neuron Ranking provides us with a list of neurons in order of decreasing importance

Algorithm 1 Neuron Ranking Extraction Algorithm

$ordering \leftarrow []$ ▷ $ordering$ will store the neurons in order of decreasing importance
for $p = 1$ **to** 100 **by** α **do** ▷ p is the percentage of the weight mass. We start with a very small value and incrementally move towards 100%.
    $tnpt \leftarrow \text{GETTOPNEURONSPERTAG}(\theta, p)$ ▷ $tnpt$ contains the top neurons per tag using the threshold p
    $topNeurons \leftarrow \bigcup_{i=1}^{L} tnpt_i$
    $newNeurons \leftarrow topNeurons \setminus ordering$
    $ordering.append(newNeurons)$
end for
return $ordering$

Dalvi et al. (2018) continued

Elastic Net Regularization

- Tune λ_1 and λ_2 - find a balance between individual neurons vs groups while maintaining the same accuracy as the original
  a) Higher λ_1 increases sparsity
  b) Higher λ_2 increases the likelihood that neurons activated by similar features will be similarly ranked

Dalvi et al. (2018) continued

Evaluation of Neurons for Linguistic Correlation Analysis

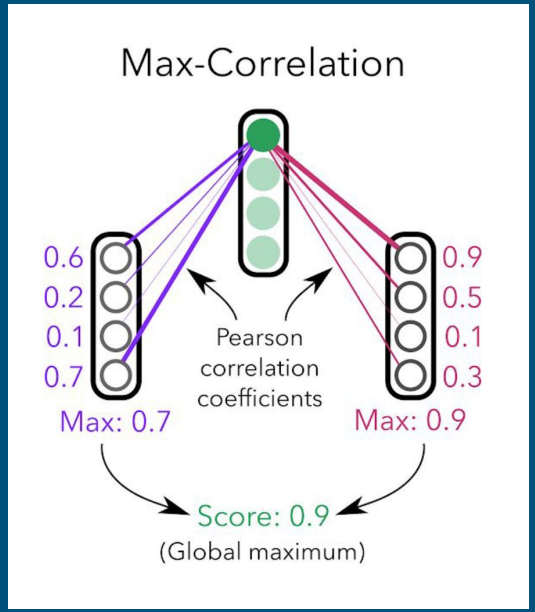
-All *except* the top/bottom N% of neurons have been masked

	Task	ALL	Masking-out					
			10%		15%		20%	
			Top	Bot	Top	Bot	Top	Bot
NMT	FR (POS)	93.2	63.2	23.8	73.0	24.8	79.4	24.9
	EN (POS)	93.5	69.8	15.8	78.3	17.9	84.1	21.5
	EN (SEM)	90.1	51.5	16.3	65.3	18.9	74.2	20.7
	DE (POS)	93.6	65.9	15.7	78.0	15.6	88.2	15.7
NLM	FR (POS)	92.4	41.6	23.8	53.6	23.8	59.6	24.0
	EN (POS)	92.9	54.2	18.4	66.1	20.4	72.4	24.7
	EN (SEM)	86.0	49.7	21.9	56.8	22.3	65.2	25.1
	DE (POS)	92.3	39.7	16.7	51.7	16.7	67.2	16.9

Dalvi et al. (2018) continued

Cross-Model Analysis (CMA)

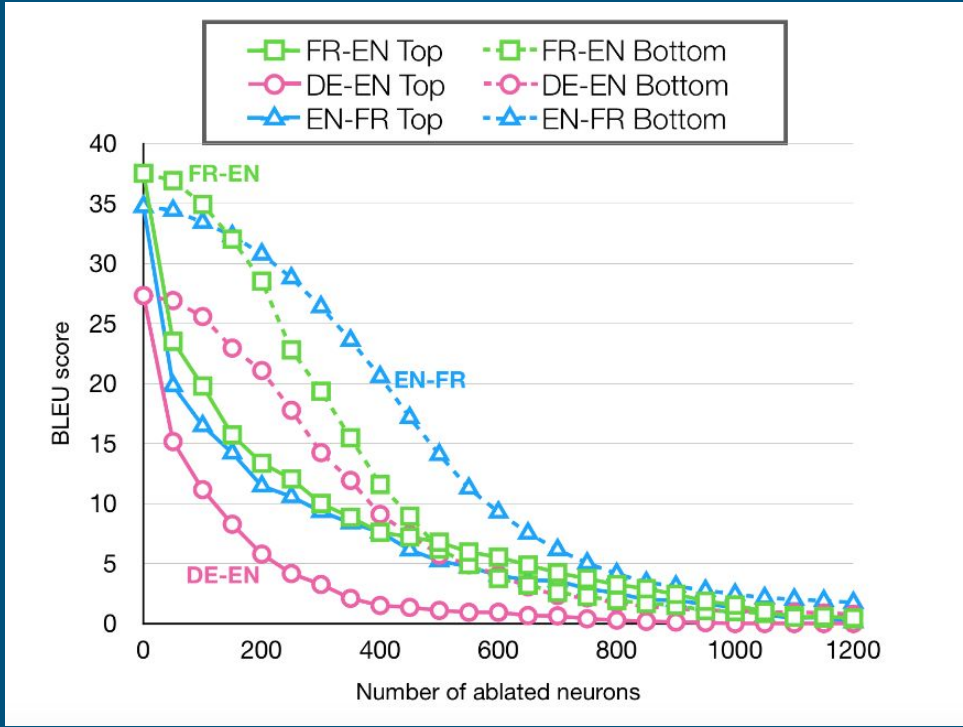
1) Run several related models
   a) Only training data and initialization is different
2) Find neurons in model M_1 which have the highest correlation to any other model (M_2,...M_N)



Dalvi et al. (2018) continued

Evaluation of Neurons for Cross-Model Analysis

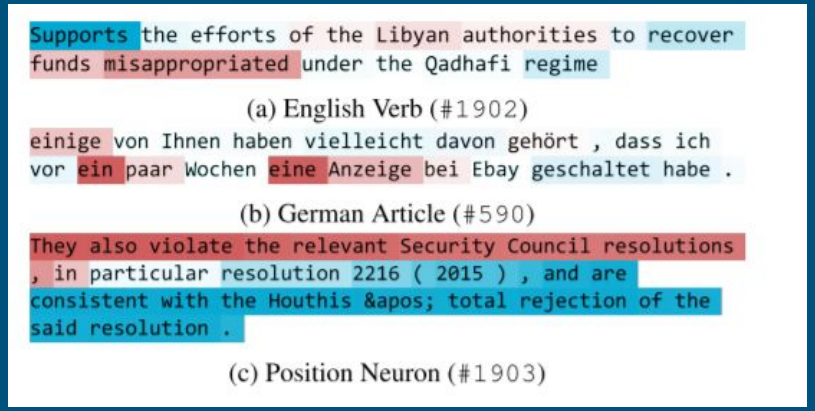
- Neuron Ablation: mask the neuron activations in the test set
  - The most correlated neurons (Top)
  - The least correlated neurons (Bottom)

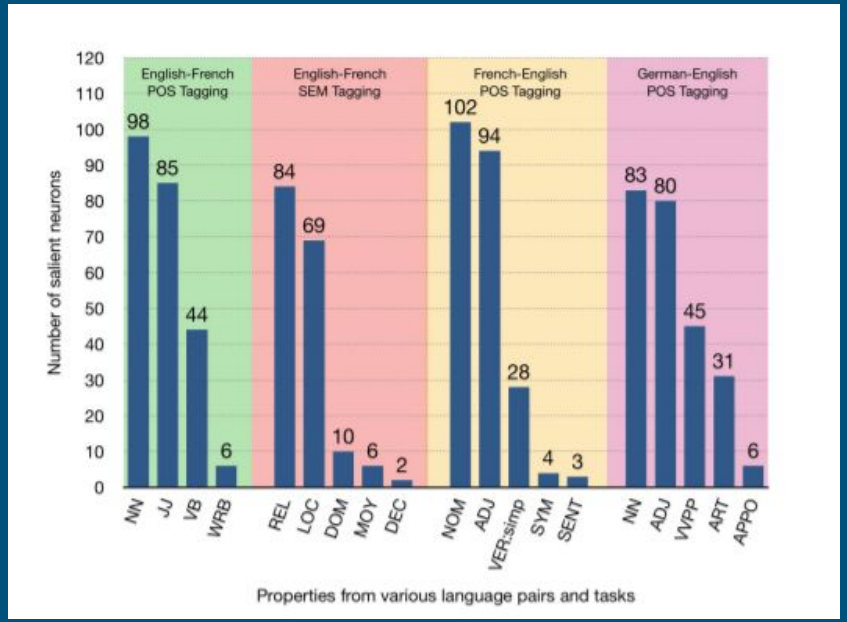


Dalvi et al. (2018) continued

Neurons and Linguistic Properties

Visualization of individual neurons in Linguistic Correlation Analysis





Number of salient neurons uncovered per each tag

Dalvi et al. (2018) continued

Conclusion

- There are both supervised and unsupervised methods to uncovering meaningful neurons
- Linguistic correlation is an especially helpful method to understand the neurons that encode specific linguistic properties
- Give it a try at: https://github.com/fdalvi/neurox