Embedding Bias: Analyzing the Semantics of Bias in Breitbart with Word Vectors

A CS221 Research Report by

Jake Rachleff, Frits van Paasschen, Luigi Sambuy
{jakerach, fritsvp, lsambuy}@cs.stanford.edu

***Abstract*—In this project, we aim to understand the difference in semantic meaning between two sets of text - one with heavy bias and one without. We build high quality word embeddings based on a biased corpus of text, and compare them with the oracle model of Facebook's Fasttext. Then, we build on word embeddings to present a novel technique for quantifying bias in text. Using our novel method, we identify the words that carry the largest difference in meaning under our new classification, and qualitatively discuss the results through charts and visualizations of the embeddings spaces.**

I. Introduction

Fake News and Media Bias have become a pressing issue in the world today. It is widely postulated that the American perception of current events and social issues has been changed by a rise of partisanship in the media. It is widely discussed that the United States is under rising partisanship and its citizens choose media sources close to their own opinion. However, as far as we can tell, no studies have looked at the *meaning of the words themselves* in biased stories through a mathematical framework.

In light of this, we take advantage of a now common idea in Natural Language Processing called word embeddings to propose a novel technique in understanding the difference in meaning in words in biased and non biased corpora. Thus, our project has two steps. We first map the semantic meaning of words from Breitbart and words from Wikipedia into vectors that can be mathematically manipulated to give understanding of their meaning. Then, we must use our novel approach to show the difference semantic meaning of biased words from Breitbart and their unbiased counterparts from Wikipedia. We hope to analyze the difference in these sets of vectors, and show that the difference in their underlying meaning corresponds to true bias in the source text.

II. Background and Literature Review

In order to better understand the methods used in this paper, it is important to first gain a solid understanding of word embeddings and embedding space projections.

A. Word Embeddings

The idea of word embeddings draws on linguist Zellig Harris' observation in 1954 that every word is defined by its context [6]. Modern word embeddings, pioneered by Mikolov et. al in 2013, map English words into mathematical space by minimizing a known cost function under a supervised learning algorithm [3]. Using this cost function, each word is transformed into a d dimensional vector, and the dot product of two vectors is high if words are similar and low if they are not. These embeddings can be mathematically manipulated to show similarity between words, perform word relations tests, and visualize meaning in low dimensional plots. The most widely accepted cost function for creating word vectors is the skipgram model.

In skipgram, we aim to minimize the cost of predicting context words given a centered word. Specifically, if we have a center word w_c, we want to predict all the surrounding values of w_t. Thus, the cost function of our supervised task is:

$$\sum_{t=1}^{T} \sum_{c \in C_t} \log p(w_c | w_t)$$

To understand this better, we should look at an example. For the sentence "I am very hungry tonight" and the center word "very", we hope to predict the context of ["I", "am", "hungry", "tonight"].

B. Fasttext

In 2016, Facebook's research division proposed a new, more advanced way to create embeddings called Fasttext [1]. This has been recognized as a cornerstone, state-of-art approach to capture word meanings efficiently. Fasttext operates by incorporating character *n*-grams into a skipgram model. Instead of just performing skipgram on words, it performs it on *n*-grams. This allows for a more granular encapsulation of the meaning of words and an enhanced performance on words that the model did not train on, since the model takes into account subword information and therefore the internal structure of words. To bound memory usage, the model uses a hashing function to map *n*-grams to integers 1 to *K*, using the Fowler-Noll-Vo hashing function[1].

As the top performing model on the standard NLP word relation dataset developed by Mikolov et. al. in 2013 [3], Fasttext's Wikipedia serves as our Oracle model on word relation performance. The test dataset consists of semantic word relations (e.g. brother : sister :: dad : mom) and syntactic relations (e.g. play : played :: run : ran). Fasttext has a syntactic performance of 70.1 % and semantic performance of 78.5 %.

C. Multi-lingual projection

Faruqui and Dyer's "Improving Vector Space Word Representations Using Multilingual Correlation" argues that lexico-semantic meaning should be invariant across different languages [2]. The authors validate their findings using canonical correlation analysis (CCA) for incorporating multilingual evidence into vectors generated monolingually. CCA measures the linear relationship between two multidimensional variables. It then finds two new projection vectors, one for each variable, which are optimized with respect to their correlations. Their methods suggest that it is possible to take two sets of semantically similar embeddings trained in different spaces, and project them into the same space.

D. Applications to Our Research

Using Fasttext, we can efficiently train high performance embeddings on the Breitbart corpus. We can then use the dataset created by Mikolov et al to analyze the semantic validity of said embeddings.

Faruqui and Dyer's work suggests that if two lists of embeddings encode the same semantic concept, then there exists some linear transformation that maps the first list onto the second. If this method works for embeddings across languages, where words do not directly translate, then there is a high likelihood that this method will work for creating a linear transformation a list of embeddings trained in English on one corpus to a list of embeddings trained in English on another corpus. We build on their work, and propose a novel algorithm for identifying and performing this transformation. Thus, we can use this algorithm to project Breitbart trained embeddings into Wikipedia space, and compare their semantic meaning.

III. Dataset

Our biased dataset comes from a known biased news source Breitbart. We have 195,875 articles from 2110 unique authors spanning a decade. For each article, along with its text, we have the date, author, url, category, and title. All told, our dataset takes up 600 Megabytes of space, 500 of which is raw article content.

Our unbiased dataset comes from Wikipedia. We first downloaded all English Wikipedia entries in XML format, and cleaned and converted this XML to nearly 14 Gigabytes of raw text, containing around 4 billion total words. However, this data was still not clean enough to generate useful metrics on performance and comparisons to our Breitbart vectors. As such, we abandoned the idea of training our own Wikipedia vectors in favor of using Fasttext's vectors trained on Wikipedia, as their data cleansing was higher quality than ours.

IV. Measuring Individual Embeddings

In this section, we describe our methodology for obtaining and evaluating embeddings trained on Breitbart. Further, we describe both quantitative and qualitative results of our evaluation. To understand the inherent quality of these embeddings in capturing the English language, we look at their performance on classic English semantic word relation tests. To understand biases baked into the embeddings, we look at nearest neighbors for words that potentially carry high amounts of bias, and compare the results for the biased and unbiased corpora. Finally, to understand the global structure of embeddings and their underlying relationships, we visualize both datasets using t-distributed stochastic nearest neighbors (commonly referred to as t-SNE) [5], and perform a qualitative analysis of the differences in datasets.

A. Obtaining Embeddings

In order to obtain the necessary word embeddings, we sourced 300 dimensional vectors based on Wikipedia from the Fasttext project. Breitbart embeddings were created running Fasttext with mostly default parameters. We modified embedding size to be 300 so as to match the dimensionality of the Wikipedia embeddings. This choice was made to preserve information during the projection made in the next section.

B. Semantic Validity

Methods: In the word relation set described earlier, we try to predict the fourth word in the relation using cosine similarity as shown in [4]. For a relation test with the format a : b :: c : d (where d is our unknown word), the prediction formula is as follows, where x_a, x_b, x_c represent the word vectors of their respective words:

$$d = \underset{i}{\operatorname{argmax}} \frac{(x_b - x_a + x_c)^T x_i}{||x_b - x_a + x_c||}$$

To give a concrete example of an input/output pair within the context of our project, we present the analogy "boy : girl :: brother : ?", where the correct answer is "sister". We tested the validity of our newly trained word vectors on the analogy tests developed by Mikolov et. al. in 2013 [3].

Results:

The results of our different embeddings can be seen in Table I. We used the score on the relation test as a marker for how well our word vectors captured English's semantic meaning. Breitbart embeddings were trained from scratch.

TABLE I: Table 1: Loss of transformation methods on fine tuned embeddings

Model	Semantic Score	Syntactic Score
Baseline	.044	.592
Cleaned Model	.287	.550
Oracle	.701	.785

We improved our model through several data cleansing techniques. Using case-matching, discarding non-alphanumeric characters, and throwing out infrequent words, we improved our semantic score from 4.4% to 28.7%. However, our syntactic score decreased to 55.0%. We believe this is due to the ngram based structure of Fasttext. When our word vectors captured very little semantic meaning, they still could understand that the ngram "ing" or "ed" corresponded to text, so they

were fantastic at guessing word relations of the form ”play” : ”playing” :: ”jump” : ”jumping”. The algorithm did not have a strong understanding of the difference between ”play” and ”jump”, so the only difference it really could understand was the ”ing”, and it was successful at appending ”ing” to jumping. Many syntactic tasks have this added suffix or added prefix form, which led to our algorithms high syntactic performance. We view this drop in syntactic score as a positive finding, however, since our interest in words' inherent bias makes us much more focused on the ability for our word embeddings to capture semantic meaning rather than syntactic meaning.

We wanted to further understand why our model's semantic score was relatively low to state of the art, and if the score on the semantic tests truly represented the word vector's understanding. Therefore, we broke down semantic analogy tests into five sections, and defined the ”closeness” score as whether our model had the correct answer within its ten best guesses. For a breakdown of our scores, see Table II.

TABLE II: Breakdown of Semantic Score on Breitbart Embeddings

Category	True Semantic Score	Closeness Semantic Score
All Semantic Categories	.287	.550
Family Relations	.453	.800
Currency	.011	.029
Less-known International Capitals	.243	.461
Popular International Capitals	.632	.844
State Capitals	.361	.785

Observe that in the Family Relation, Popular Capitals, and State Capitals sections, our embeddings perform much better than their broad semantic score. Also, notice that ”closeness” scores are often 20-30 % higher than their counterpart. The largest section of all was the Less-known International Capitals section (4524 examples), which was almost nine times bigger than Family (506 examples). Together, this suggests three things. First, Breitbart is anti-globalist, which may lead to lower scores in globalist tasks like identifying foreign currency or less popular foreign capitals. Second, the fact that closeness score is so much higher shows that the embeddings do capture more semantic meaning than their score initially suggests. Finally, the word relations test set was built for Wikipedia, so heavy weight was put on globalist sections that Wikipedia vectors would perform well on, but Breitbart vectors do not, because they do not discuss most global affairs. Thus, we believe these embeddings would perform better on a more balanced dataset, and are confident in their ability to capture semantic meaning.

C. Nearest Neighbors

In order to show that word embeddings generated on different corpora can encode different meaning for the same word, we ran a nearest-neighbors test on both Breitbart and Wikipedia-generated word embeddings.

Methods: We find the top ten nearest neighbors X' of a word X by using the cosine similarity function as follows:

$$X' = \underset{Y}{\operatorname{argmax}} \frac{X \cdot Y}{||X|| * ||Y||}$$

In finding the top ten words X' such that the cosine similarity is maximized between X' and the target word X, Fasttext gives insight into the qualitative difference in meaning of individual words between Breitbart and Wikipedia.

Results:

TABLE III: Nearest Neighbors Analysis for Key Words

Word	Nearest Neighbors Wikipedia	Nearest Neighbors Breitbart
immigrant	immigrants, immigranted, immigrants-took, emigrant/immigrant, emmigrant, immigranten, immigrating, immigrates, immigrations, 'immigration'	immigrants, illegal, non-immigrant, immigrate, undocumented, immigrations, immigrated, illegals, nonimmigrant, alien
black	white, black/black, -black, # black, gray/black, $\sim$ black, w/black, white/black, blue/black, black&white	african-american, african-americans, blacks, hispanic, panther, mexican-american, color, black-on-black, hoodie, asian-american
snowflake	snowflakes, snowflakers, snowflaking, antisnowflake, snowflakess, snowflower, snowflex, wflake, wolflake, snowfolk	snowflakes, snowfall, f*g, sjw, geek, snow, milo, yiannopouloss, nerd, yiannopoulos
however	although, but, nevertheless, though, nonetheless, 'however', unfortunately, >however, ultimately, consequently	although, though, but, that, nonetheless, nevertheless, indeed, also, fact, noted

We chose to run a nearest-neighbors comparison test on both Wikipedia and Breitbart-generated embeddings using several words that are likely to have shifted meanings due to bias. The results of this test can be seen in Table III. Words such as 'immigrant' are similar to words of the exact same, or opposite meaning in Wikipedia space. However, in Breitbart space, the meaning of the word 'immigrant' has been changed to include topics relating specifically to illegal immigration. This is also seen, perhaps more drastically, with words such as 'black' (which is grouped with other minority groups and racial biases in Breitbart), and 'snowflake' (which is grouped with popular alt-right insults and slurs). These results suggest that words trained on Breitbart contain the alt right associations of the original text. This serves as an important piece of evidence in our hypothesis that bias is baked into Breitbart embeddings.

D. Individual Embedding Visualization and Analysis

Methods: We used t-SNE to visualize both Breitbart and Wikipedia embeddings. The specifics of the algorithm behind

t-SNE is beyond the scope of the course, and can be found in [5]. At a high level, t-SNE does not need to preserve linear structure. Instead, it aims to best describe local clusters. It models conditional probabilities of closeness through pairwise distance of points in two dimensions, and performs gradient descent to find optimal clustering. Depending on the value of the hyper parameter perplexity, the graph can focus on global or local structure. We plot the top 500 most common words in the Breitbart corpus with length of at least 5 to weed out articles and other meaningless words at a variety perplexity values and visualize the results of the most comprehensible graphs.

Results:

The full visualizations created using t-SNE are too large to include in their entirety, and are left to the interested reader in the appendix. Instead, we zoom in on pieces of each graph to show how embeddings cluster together and share similar roles and meaning in both Breitbart and Wikipedia space.

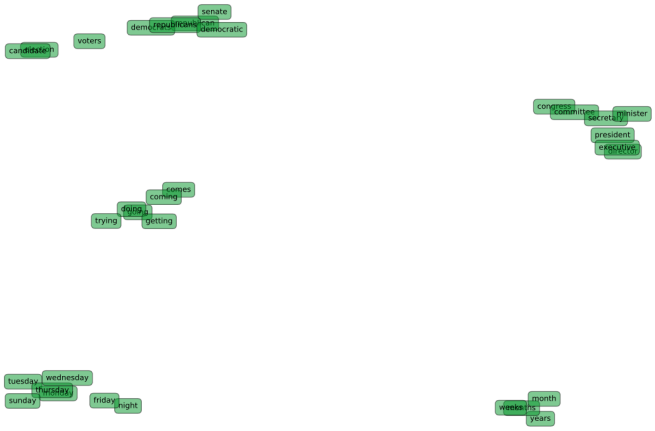

Fig. 1: Clear clustering of similar embeddings in Wikipedia embedding space

First, let us consider the Wikipedia Vectors. We chose a low perplexity value ($p = 1$) to allow for more local structure. Examining Figure 1, notice that similar concepts group together. The bottom left is days of the week, the bottom right is time periods, the top left are political groups, and the top right are political positions. This helps exemplify the power of embeddings — words with similar meanings and contexts are close in value.

Now, let us look at two specific clusters. In Figure 2 we see the representation of universities on the right, as well as economic and national issues on the bottom, and Middle Eastern Countries in the top right.

In Figure 3, we see the representation of force clusters. Notice that military and force are grouped together, and terrorist and attacks are grouped together. Each mini cluster in this figure is close together because of huge overlapping meaning and context, and the mini clusters are in the same overarching cluster because they both have to do with war.

Now that we have a clear sense of the layout of embeddings in Wikipedia space, we can move on to those in Breitbart space. We first examine a larger set of clusters in Figure 4.

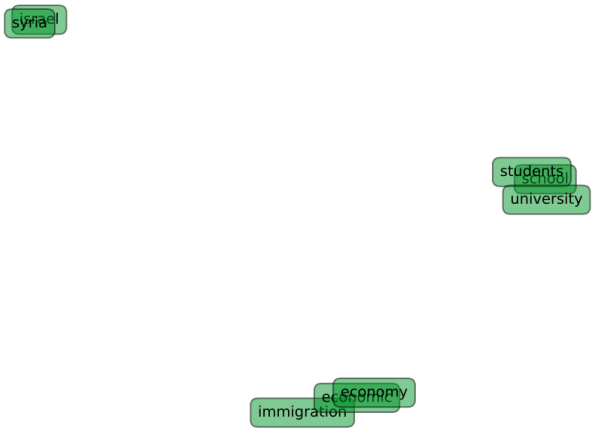

Fig. 2: Specific topics in Wikipedia embedding space

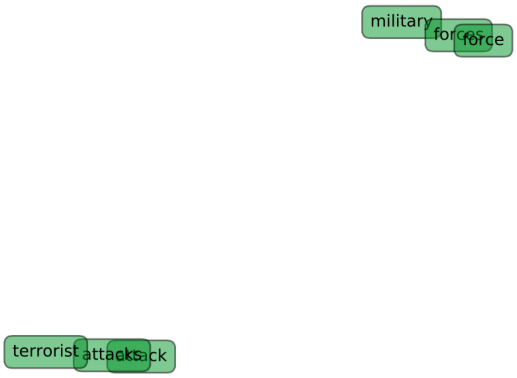

Fig. 3: Force cluster in Wikipedia embedding space

Notice that these embeddings clearly hold semantic structure. The top right cluster contains words used for interviews and debates, bottom right contains ”force” words, top left has the association of America and country. Now, focusing on the bottom left, we start to see the difference in structure from Wikipedia. The bottom left cluster has immigration and illegal side by side, since in Breitbart, immigration is usually only talked about in the context of the border with Mexico. This differs from Wikipedia, where immigration was placed next to economy, since both are issues of national policy.

This effect can be seen even more clearly in Figure 5. Here, we see two clusters, one in the bottom right representing terms for leaders in groups that would make sense in any English context, and one in the top right that is far more interesting. We contrast the two to illustrate that this placement is not accidental. Notice that the force cluster from Wikipedia space that contained attacks and terrorist now is placed next to muslim and islamic. In articles in Breitbart, Islam and Terrorism are closely intertwined. Here, we see a perfect example of word embeddings capturing the bias baked into Breitbart.

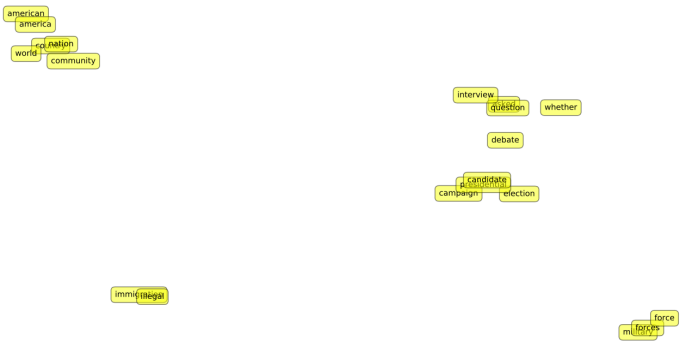

Fig. 4: Clear clustering of similar embeddings in Breitbart embedding space

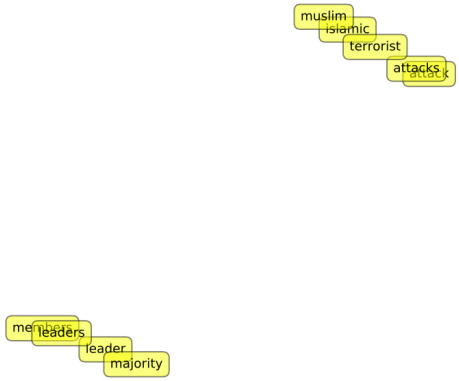

Fig. 5: Semantic similarity of Muslim and Terrorist in Breitbart

V. Bias Measurement with Embedding Space Projections

Examining the individual visualizations in the above section, along with the full visualizations in the appendix, a clear pattern emerges. Highly politicized words like "muslim" and "immigration" tend to change meaning in a biased corpus, whereas apolitical words do not. Though this may be a useful observation, it takes large amounts of time and judgment to determine which words appear to be most different in each space. Further, there is no quantitative metric that determines which words have the most different semantic meaning between spaces.

In this section, we propose a novel method to quantitatively identify words in dataset that carry the strongest amount of bias. This method is entirely unsupervised and, as long as there exists some unbiased set of vectors (i.e. Wikipedia), provides high quality results.

A. Cross-Space Embedding Comparison

Visualizations in the previous section suggest that distance is a natural measurement for difference in meaning between two embeddings in the same space. However, several issues arise when measuring distance across two different embedding spaces. First, the average vector for each space may not be at the origin, which means that in order to compare properly, we must zero center each dataset. Second, Fasttext does not normalize vectors, so each set of embeddings may be at a different scale. Thus, we also must divide each dataset by their respective standard deviations in each dimension in order to have normalized and comparable data.

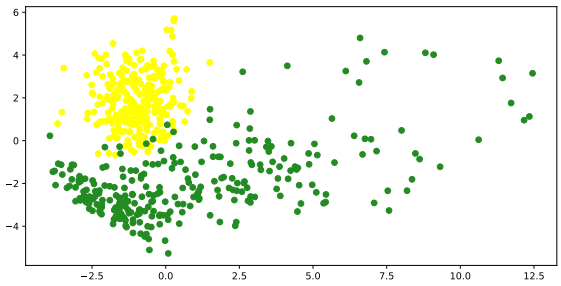

Fig. 6: Zero Centered and Normalized Breitbart and Wikipedia Embeddings

At this point, it may seem like the data is ready to compare. In Figure 6, the yellow dots represent embeddings in Breitbart, and the green dots represent embeddings in Wikipedia. Though we have zero centered and normalized both datasets, they still appear to be entirely separate entities with small overlap.

Consider a single corpus run through Fasttext twice. Fasttext optimizes using stochastic gradient descent on the skipgram cost function (plus subword information), which means that it is not guaranteed to find a global minimum. Thus, the solution it finds for a single dataset is not unique, and several sets of final vectors could bring similar loss. These vectors should contain similar semantic structure, meaning they are in the same locations in space relative to each other. They could be all be scaled or rotated, since these linear operations should not affect the embeddings' relationships relative to one another. Simply finding a projection that properly scales and rotates the embeddings should put them in the same space, and make them easily comparable.

Now, consider two separate corpora run through Fasttext where the majority of the elements should have similar meaning. If elements in the first corpus have the same meaning as in the second corpus, they should each map to the second corpus with a single rotate and scale operation. Thus, if the majority of elements have the same meaning in the first corpus as the second, there exists some projection that maps the majority of elements in the first corpus to the space of the second corpus. Projected elements from the first space should be in close proximity to their counterparts in the second space if they carry similar meaning in both spaces, and far away if not.

This situation is exactly what happens with the Breitbart and Wikipedia corpora. The majority of elements have the same meaning in both corpora, but a select few have wildly different meaning. Thus, there should exist some projection that rotates and scales non biased embeddings from the first space into the second. Words that are biased in Breitbart should be far away from their counterparts in the second space, since their relative meaning has been warped.

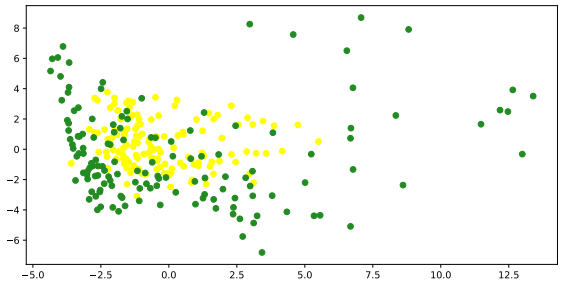

Fig. 7: Breitbart embeddings projected into Wikipedia space

In Figure 7, we see the effect of performing a projection on the Breitbart embeddings. The two sets appear to be similar, except for values on the far right of the plot (which happen to correspond to political terms). In the next section, we describe our methodology for finding such a projection.

B. Projection Methods

Below, we outline a methodology to properly project Breitbart vectors into the Wikipedia vector space, then use the projected embeddings compare the difference in Breitbart word embeddings and Wikipedia word embeddings for the same set of words.

1) Obtaining the Projection Matrix: Given previous work, we believe that if word vectors can be projected into another language's vector space because they contain the same meaning, then English word vectors trained on one dataset should be able to be projected into the space of English word vectors trained on another dataset.

To test our projections, we produced two sets of word embeddings trained from scratch — one based on Wikipedia and the other based on Breitbart. We aim to use the methodology outlined below to find projections from Breitbart space to Wikipedia space.

If both sets of embeddings have dimension n, then a projection from the first space to the second can be defined as an $n \times n$ matrix. Say we have a matrix of row vectors X_1 that we want to project into another space to be as close as possible to another matrix of row vectors X_2 (where X_1 and X_2 might represent embeddings in Breitbart space and Wikipedia space respectively). To find an optimal projection, we aim to find the best matrix P such that $X_1 P \approx X_2$.

We run three experiments to find such a matrix. First, we run the control experiment by setting $P = I$. This allows us to compare our other projection methods and see if we can find a better fit. Second, we use linear regression to estimate the optimal projection matrix. Recall that given the equation $X_1 P = X_2$ where we know X_1 and X_2, linear regression estimates P as follows:

$$P = (X_1^T X_1)^{-1} X_1^T X_2 \tag{1}$$

Third, we use a modified linear regression algorithm to estimate P. Vanilla linear regression operates on the assumption that all points should be taken into account when minimizing the error of our projection. However, biased words are expected to not be close to their unbiased counterparts post projection. Thus, we should only select certain key points in our corpora that likely carry less bias, and transform based on them. Thus, for our third projection estimation, we perform the algorithm defined in Algorithm 1 to find the most similar words in two vector spaces, and only project based on those words.

Algorithm 1 Residual Tossing Linear Regression

procedure RESTOSSLINEARREGRESSION(X_1, X_2, k)
  bestProjection ← LinearRegression(X_1, X_2)
  while Length(X_1) $> k$ **do**
    $residualIndeces$ ← MinResidualIndeces(X_2)
    X_1, X_2 ← FilterIndeces(X_1, X_2, residualIndeces)
    bestProjection ← LinearRegression(X_1, X_2)
  return *bestProjection*

At each iteration, the algorithm performs linear regression and finds the indeces of X_1, X_2 that led to the largest residual error. At every iteration, it filters every word vector that caused a residual larger than the median residual. Finally, if the number of vectors contributing to the linear regression is less than k (which we set to 2000), we return the projection, since we know this projection should not have been affected by outliers.

2) Using Projected Vectors to Measure Bias: Once the projection matrix is obtained, we first find the set of vectors that appear in both Breitbart and in Wikipedia. We then use our projection matrix (P) to project all of said vectors in Breitbart space (X) onto Wikipedia space to from a new set of vectors (X'). We can then define the distance between a word embedding x in X and its equivalent x' in X' as follows:

$$d = \sqrt{\sum_{i=1}^{n} ((xP)_i - x'_i)^2} \tag{2}$$

This is also known as the l_2 norm of the difference between the two vectors.

C. Measuring Bias in Words

These projection metrics are the framework needed for finding words with the biggest change in meaning in the Breitbart corpus. Given the definition of bias as difference in meaning, we can use our distance equation defined above to rank bias amongst words in our dataset for each of our projection methods. To deeper understand our output, we list the top 25 most biased words above different frequency thresholds. First, we look at only the 800 most common words in Breitbart that are longer than 4 letters (to weed out junk ngrams). These results can be seen in Table IV.

These results are promising, in that most "biased" words found by our methods can definitively be connected to a topic of bias. For example, words such as "border", "syrian", "refugees", "immigrants", and "amnesty" are all clearly related to issues of immigration. In addition, "email" most likely refers to the Hillary Clinton's email scandal during the 2016

TABLE IV: Top 25 Most Biased Words with Zero-Centering and STDev-Normalization: Filtered by 800 Most Frequent Words

No Projection	Least Squares	Residual Tossing
debate	debate	debate
amendment	comments	comments
county	players	border
prime	class	amendment
senate	obamas	trumps
abortion	county	county
refugees	families	amnesty
nuclear	trumps	trump
players	trump	governor
border	border	billion
comments	added	chairman
israeli	breitbart	prime
rubio	report	class
chairman	amendment	syrian
minister	prime	email
syrian	americas	players
islam	media	federal
class	district	added
families	federal	media
email	further	youre
congressional	governor	union
district	department	obamacare
obamacare	donald	nuclear
immigrants	liberal	liberal
afghanistan	romney	secretary

TABLE V: Top 25 Most Biased Words with Zero-Centering and STDev-Normalization: Filtered by 8000 Most Frequent Words

No Projection	Least Squares	Residual Tossing
rebounds	households	sinai
touchdowns	debate	melgen
weekdays	bidens	schweizer
touchdown	subsequent	coptic
households	census	siriusxm
tsipras	latino	gruber
jinping	rebounds	households
yards	observatory	medal
innings	males	guantanamo
medal	stadium	nobel
census	recipients	mount
yazidi	scores	tayyip
siriusxm	nolink	keystone
tayyip	clubs	debate
inning	gorka	ballots
fetal	ivanka	voiced
recep	beltway	duterte
olympics	domain	jinping
melgen	news	melania
seahawks	comments	emanuel
yazidis	boxes	subsequent
hurricane	stakes	subcommittee
lakers	ratio	peninsula
subcommittee	register	devos
latino	please	rigged

Presidential Election. Finally, ”obamacare” clearly points to a certain bias towards health care policy, and specifically the health care policy implemented by the Affordable Care Act. Not only do these examples show the validity of our methods, but they allow us to begin the process of qualitatively analyzing what types of bias are present in our Breitbart corpus.

However, as encouraging as these results may seem, they also show clear areas in which more insight is needed about our projection methods. As is evident from Table IV, we observe similar performance across all projection methods. This is in contrast to what we would expect; an identity projection should not be as accurate at capturing biased words as a winnowing algorithm that fits the data in a more nuanced way. Nonetheless, there are several shared words between the most biased set under an identity projection and the most biased set under residual tossing. We found that when increasing the threshold for word frequency, the consistency of our identity-based projection broke down, while our residual tossing based projection accurately captured people and topics with implied bias. These results can be seen in Table V, where we explore the top 8000 most frequent words in Breitbart.

Here, we see that the implied bias most of the farthest 25 words under our identity projection is much less clear than the other tests. For example, we characterize seemingly apolitical words such as ”yards”, ”innings”, ”medal”, and ”recep” as high bias. This suggests that that this method introduces error into our results as the frequency threshold is raised. Many of these errors contain sports terminology, which may help explain why the unprojected test performs well on small datasets but not large ones. If a few embeddings are highly biased and have much different values than their corresponding Wikipedia embeddings, then in spite of their orientation they will still exhibit high bias, since they are still most likely in a separate part of the embedding space. Referring back to Figure 6 which was created without a projection, no matter which yellow Breitbart point corresponds to the farthest right green Wikipedia point, the corresponding word will register as high bias since all yellow Breitbart points are far away.

However, as more terms are introduced, we may find that the lack of orientation alignment may lead a certain cluster of projected embeddings to be located far away from the same cluster of unprojected embeddings by raw chance. Visually, this would correspond in Figure 6 to the Breitbart representation of sports being on the top side of the yellow cluster, and the corresponding representations in Wikipedia being on the bottom side of the green cluster. This would help explain why 11 of the top 25 most biased words were registered as sports words.

Though the Least Squares algorithm may seem high quality at first glance, it tends to overfit because it takes into account biased and non biased words. This could be the reason we see words like ”rebounds”, ”scores”, ”clubs”, ”boxes”, and ”ratio” that are difficult to qualitatively explain as high bias.

In contrast, our residual tossing algorithm is able to accurately predict more refined topics with clear implied bias. The word ”schweizer” refers to Peter Schweizer, an investigative journalist known for writing about the Clinton Foundation, ”Sinai” and ”Coptic” refer to areas in Egypt where Breitbart reports that Muslims kill Christians, and ”keystone” refers to the Keystone XL pipeline, which prompted widespread debate in 2016. Each of the topics in the top 20 for Residual Tossing can be mapped to a specific hot button political scandal or world event. Overall, we observe that the results using residual-tossing are more nuanced and provide us with a more specific insight into biased topics within Breitbart.

To sanity check our projection algorithms, we also observed the closest words by l_2 distance using our different projection methods. This provides us with assurance that our projections are based on words with low implicit bias. These words are seen in Table VI. With each of our methods, we observe that the words closest by l_2 distance after being projected are qualitatively unlikely to imply any bias. As can be seen from the table, no words are present that reference topics related to any form of racial, political, gender, or economic bias.

TABLE VI: 15 Least Biased Words with Zero-Centering and STDev-Normalization: Filtered by 8000 Most Frequent Words

No Projection	Least Squares	Residual Tossing
maybe	intentionally	interestingly
suppose	similarly	surprising
course	questioning	might
having	realize	stated
whereas	might	dealing
telling	later	perceived
whatever	understand	consequently
really	worried	effectively
undoubtedly	beginning	informed
later	possibility	practically
consequently	others	there
initially	inevitable	considering
today	eventually	supposed
eventually	allowing	supposedly
reportedly	doing	embarrassing

D. Joint Embedding Visualization and Analysis

In order to fully understand the effects of our projection, we must create a visualization of our projected Breitbart embeddings and Wikipedia embeddings in the same space. Again, we use t-SNE for our visualization, and maximize local clustering with perplexity set to $p = 1$. We use our residual tossing algorithm to create the projection, and plot embeddings that are in the top 250 most used words in Breitbart, and meet the length cutoff of 4. In this section, we focus on small sections of each visualization, but the full plot is available to the interested reader in the appendix. Remember that t-SNE is non linear, so relative distances are not perfectly preserved.

In all figures in this section, please note that the projected Breitbart embeddings are yellow and marked with _proj on their label, and Wikipedia embeddings are green. First, notice in Figure 8 that apolitical words that join sentences and promote flow in English all appear in the same general cluster. This is consistent with the idea that our key points maintain closeness.

Fig. 8: Apolitical terms all appear in the same general region in the center of the plot

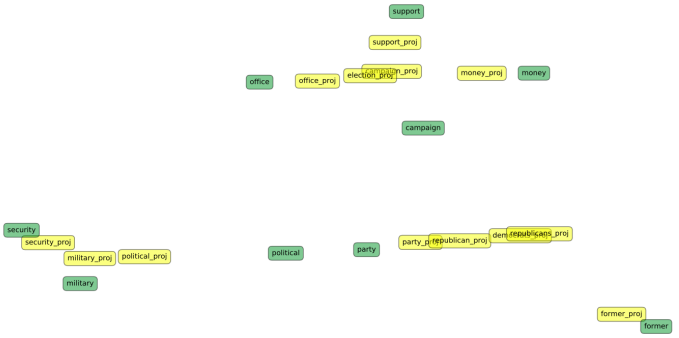

Fig. 9: Cluster of political terms

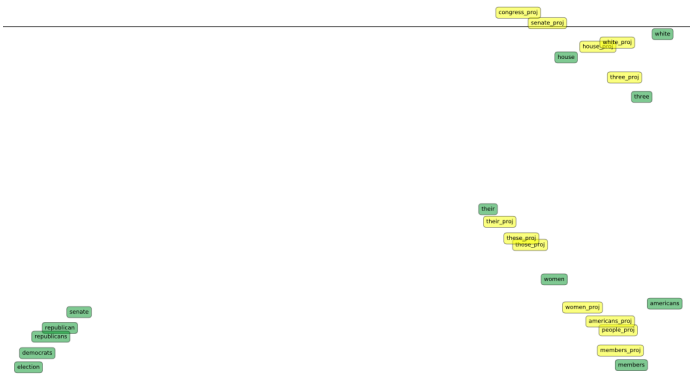

Fig. 10: Certain political terms in Wikipedia appear far from their Breitbart counterparts

In Figure 9, we see clusters where many political terms are correctly projected close to their neighbors. Four embeddings appear from Breitbart that do not have close by corresponding Wikipedia embeddings: election, democrat, republican, and republicans. We see in Figure 10 that our missing terms appear

far away in a different part of the plot, suggesting a difference in meaning in the two spaces.

Fig. 11: Presidential candidates move closer together in Breitbart

In Figure 11, a similar effect can be seen. In a cluster focused mostly on America, Obama and last year's presidential candidates appear together. All three appear close to Donald Trump's Wikipedia embeddings even though Clinton and Obama are far away in Wikipedia space. This implies that the meaning of Clinton, Obama, and Trump are all warped together in Breitbart.

Together, and along with the full chart in the appendix, these visualizations suggest that core words in English have the same meaning in Breitbart as they do in Wikipedia. However, looking at political terms like Democrat, Republican, and Clinton, we see the meaning is different in Breitbart space. This nearest neighbors layout is exactly what we'd expect with a biased political dataset where certain apolitical words overlap.

VI. Future Work

Our novel methods serve as a jumping off point for unsupervised bias analysis. Without labels, we are able to accurately assess which words carry the most bias in the Breitbart corpus. In order to further justify the validity of our model, we need to perform this analysis on corpora with varying amounts of bias across the political spectrum. Some of the bias described in Breitbart could be due to its status as a political blog rather than its political beliefs and fervent ideology. Thus, we would need to perform this analysis on multiple news sources to see which terms are uniquely biased to Breitbart, and which terms are biased across the political landscape.

Further, analysis of results from this quantitative method is purely qualitative. We would like to develop methods on top of our projection algorithm that can detect specific forms of bias, and compare results across corpora. We envision creating datasets unique to specific forms of bias (i.e. racial, gender, class, etc.), and measuring the difference in average distance between each bias set and some control set. We would perform extensive qualitative analysis to ensure the validity of these results.

VII. Conclusion

The problem of identifying bias is tricky. Until now, to do so, a person had to label data as biased or unbiased. This process inherently introduced their own views into the dataset. By providing an unsupervised algorithm to quantify and qualify bias, we present an algorithm that is not beholden to the views of a single person or organization. Thus, in a time of increasing political polarity and distrust of media, we hope to provide the groundwork for bias detection that helps people from both sides understand the veracity and centrality of their daily news intake.

References

[1] Bojanowski, P., Grave, E., Joulin, A., Mikolov T. Enriching Word Vectors with Subword Information. CoRR, abs/1607.04606 (2016)

[2] Faruqui, Manaal & Dyer, Chris. Improving Vector Space Word Representations Using Multilingual Correlation. 14th Conference of the European Chapter of the Association for Computational Linguistics 2014, EACL 2014. 462-471. 10.3115/v1/E14-1049 (2014).

[3] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Proceedings of International Conference on Learning Representations (ICLR).

[4] Mikolov, T., Wen-tau Yih, & Geoffrey Zweig (2013). Linguistic Regularities in Continuous Space Word Representation. In Proceedings of The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

[5] L.J.P. van der Maaten and G.E. Hinton. Visualizing Data Using t-SNE. Journal of Machine Learning Research 9(Nov):2579-2605 (2008).

[6] Z. Harris. Distributional structure. Word, 10(23):146–162.(1954.)

APPENDIX A
INDIVIDUAL EMBEDDING T-SNE VISUALIZATIONS

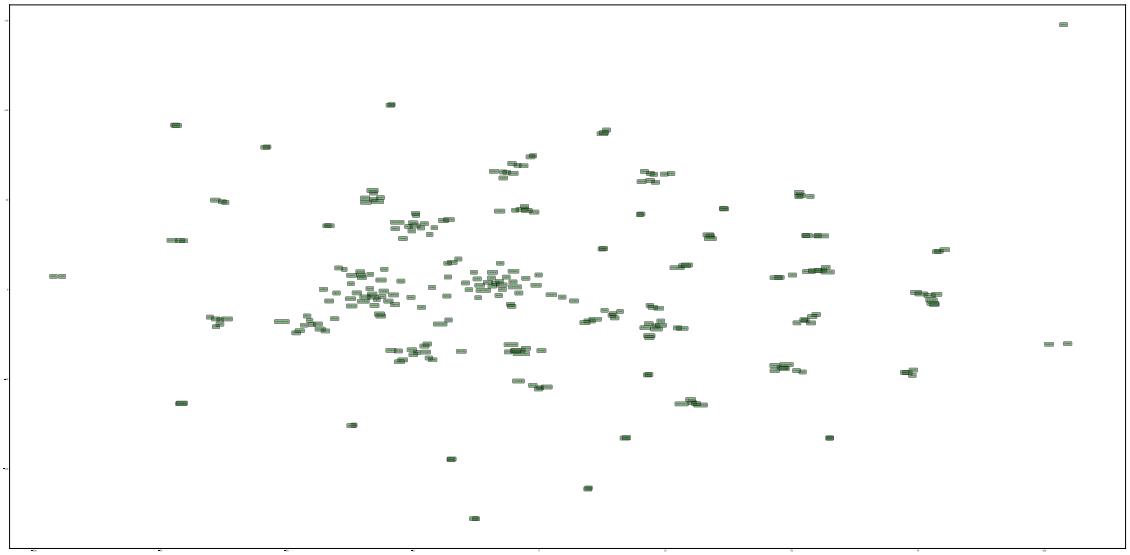

Fig. 12: Visualizing Wikipedia Embeddings in 2D via t-SNE.

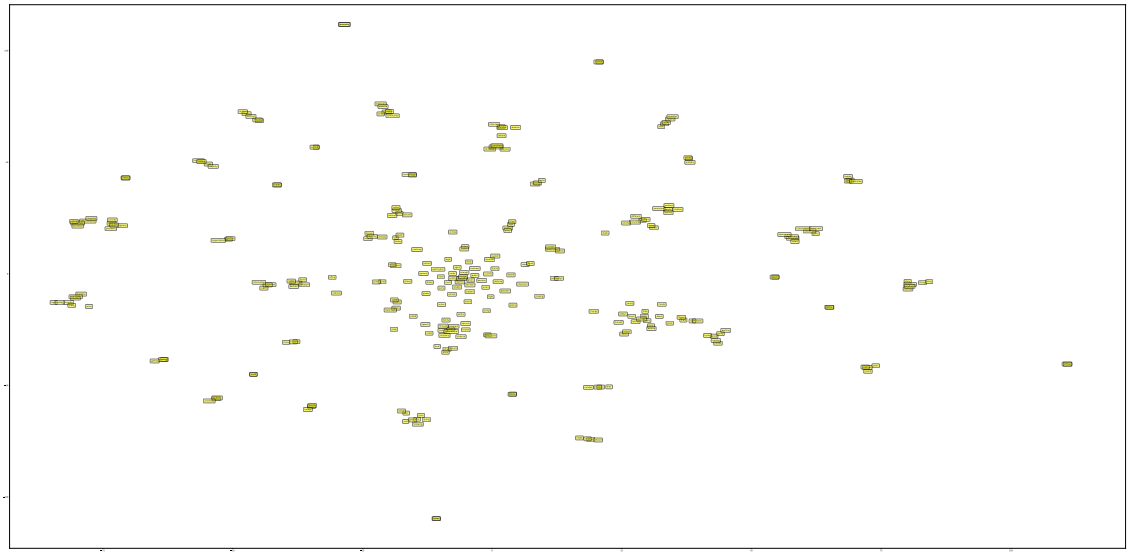

Fig. 13: Visualizing Breitbart Embeddings in 2D via t-SNE.

APPENDIX B
JOINT EMBEDDING T-SNE VISUALIZATION

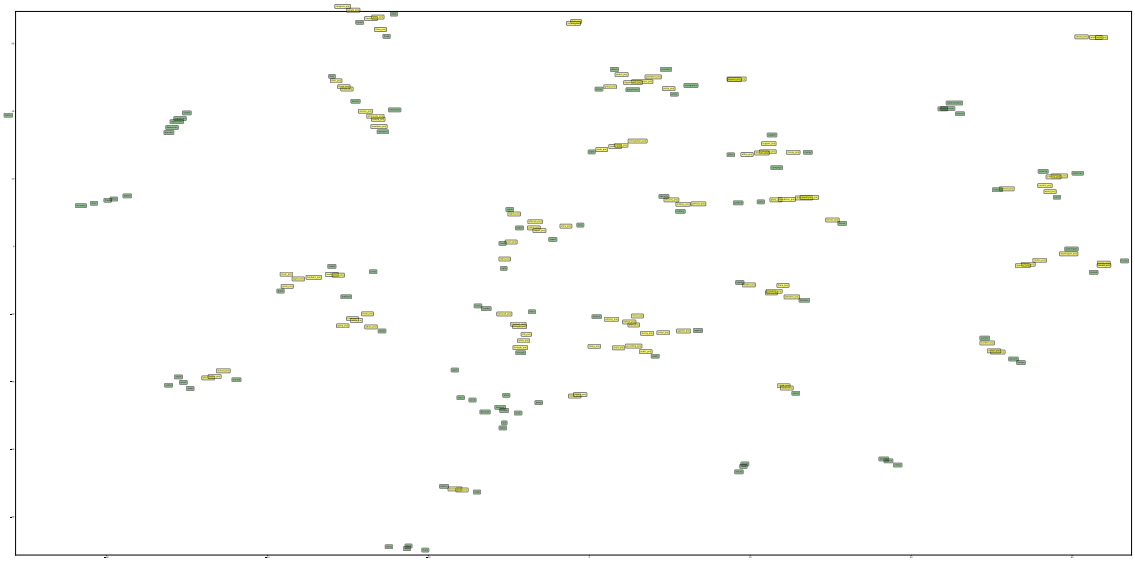

Fig. 14: Visualizing Breitbart and Wikipedia Embeddings Projected into the Same Space with Residual Tossing in 2D via t-SNE.