Stylistic Author Attribution Methods for Identifying Parody Trump Tweets

Isaac Peña
Yale University
isaac.pena@yale.edu

Abstract

Style-sensitive methods for authorial attribution have been proposed and developed in recent years in order to assess the authorship of documents with content unrelated to the rest of an author's body of work. We aim to train a classifier with a style-sensitive feature space alongside one without style sensitivity to examine how vital stylistic analysis is to a similar task - parody detection. The distinct style of Donald Trump's tweets has spun off a litany of parody tweets and parody accounts: we will train the classifier on Trump tweets and use it to determine whether tweets by other users, including non-parody accounts and parody accounts of similar and different content to Trump, are, in fact, parody.

I. Introduction

An author with an immediately recognizable style generates immediately recognizable pastiches, or, if the intents of the pastiches are sarcastic, immediately recognizable parodies. Authorship attribution is a natural language processing task that can use *both* the style and the content of an author's (attributed) body of work to determine if an anonymous document was actually written by that author [Tschuggnall and Specht, 2014]. An attempt at holistic author attribution needs to take into account non-content features in the case that the content of the training corpus is disjunct or near-disjunct from the content of the anonymous document(s) that we are trying to attribute to an author (take, for example, Nabokov's *The Real Life of Sebastian Knight* cf. the rest of his non-detective-story work). In many cases of parody, an author's style is emulated while content - settings, characters, or even themes - can change [Gamon, 2004].

In this experiment, however, the corpus used is the large and continually expanding group of stylistically similar texts published by the sitting president, Donald Trump. Sampling from his largely consistent body of work published to his Twitter account, we also drew from five other pundits from across the mainstream political spectrum who have active Twitter accounts, and chose a handful of Trump parody accounts with various political perspectives, two of which largely tweet about non-political or tangentially political subjects in 'Trump voice' ([@DungeonsDonald] and [@realNFLTrump], who post about Dungeons and Dragons and professional football respectively).

In constructing a feature space that attempted to extract and parallel stylistic features against content features, we found that many problems combined to blur the final results away from a realistic model that could reliably answer if a tweet was a parody of Trump. Some of these include the different styles of writing being used by the Twitter authors whose data trained the model, the incongruous timeframes during which the authors were tweeting (and thus different subjects of jokes and tweets), the dissimilarity of phrastic expressions across authors to refer to similar entities or events, multiple authors using Trump's account and our attempts to distinguish his wheat from his staffers' chaff, and the presence of actually

'bad' (i.e. not identifiable as a parody of Trump, not 'not funny', although the former certainly seems to imply the latter) parody tweets in the dataset by authors that do not frequently attempt to write in the authorial Twitter style Trump uses and instead tweet topical political assessments in another voice, or tweets which misuse certain aspects of Trump's style.

II. Related Work on Authorship Attribution

Plenty of research in the field of natural language processing has been done to create stylistic-sensitive (or *neustic*) feature spaces which successfully recognize an author. Tschuggnall and Specht compile various features extractable from syntactic representations of sentences in a document, summing "pq-gram" indexes and frequencies and then comparing their distributions to those of other authors using lowest distance or highest similarity metrics [Tschuggnall and Specht, 2014]. Gamon combines a shallow feature space with function word frequencies and part-of-speech trigrams with one based on context-free grammar production frequencies to reliably assign anonymous documents to one of three Bronte sisters using a support vector machine [Gamon, 2004]. Layton, Watters, and Dazeley attempt authorship attribution on tweets, and use a source code author profile method with n-grams, but clean their data first to remove hashtags and @s,[1] which sanitizes and trims certain parts of an author's style which are relevant to the medium, and do not use a support vector machine to reach their conclusions [Layton, Watters, and Dazeley, 2010]. Raghavan, Kovashka, and Mooney deploy probabilistic context-free grammars like Gamon's, but use standardized data like the Wall Street Journal and Brown corpuses and divide authorship predictions between each category of news content (football, cricket, etc) [Raghavan, Kovashka, and Mooney, 2010]. Lastly, Uzuner and Katz use a novel dimension of nested clause depth within each sentence in their feature space, but find anyway that selected function words across a (large) text perform more reliably for author attribution tasks [Uzuner and Katz, 2005].

Some of these researchers had used support vector machines for classification, and many investigated the implication that style is a 'better' determiner for attributing authorship, with differing results depending on the particular technique applied. What they had not attempted to investigate was whether 'good' parody could be recognized by the same token, and whether or not a style-insensitive feature space would be better than a style-sensitive one at distinguishing content-dissimilar parodies from non-parody tweets.

III. Data

We selected five pundits in addition to Donald Trump who actively use Twitter and tweet mostly about politics in order to keep content similar across the authors who would make up the training data. Though the specific topics and entities that a particular person considers to be "politics" might change depending on their ideological position, these pundits were chosen for their positions and the positions of their readership or followers, which largely fall in the middle of the range of American politics, i.e. from 'liberal' to 'conservative'. Regardless, these authors use Twitter to talk about "politics" often enough in the same terms as how @realDonaldTrump talks about "politics" that the content of the training data is largely content-similar.[2] These accounts are:

- @DouthatNYT [Douthat, 2017]
- @juliaioffe [Ioffe, 2017]
- @GovMikeHuckabee [Huckabee, 2017]

[1] @-ing someone at the start of a tweet indicates a conversation with that user.

[2] Some of these authors are using Trump's terminology for various entities and events in derision, others in agreement, and others perhaps because of widespread linguistic dissemination of terms that Trump often uses (e.g. "fake news"). The hold his specific language has on the realm of Twitter political discourse is palpable, and (mere speculation) perhaps stems from both his recognizable style and the tone with which media reports on his larger-than-life influence on national politics.

- @jonathanchait [Chait, 2017]
- @mattyglesias [Yglesias, 2017]

These accounts each received their own class in the support vector machine and are thus guesses in the final predictions for authorship that the classifier makes, alongside @realDonaldTrump.

Acquiring the Trump tweets themselves proved to be something of an interesting wrinkle as well - though the other five training authors tweet with their own style, the president, at times, has his staffers use a different device to tweet from his account for various promotional messages. In 2016, it was recognized by David Robinson after an observation by Todd Vaziri that the author of a tweet was probably *not* Trump if the tweet was sent by iPhone, and that the author probably *was* Trump if the tweet was sent by Android [Robinson, 2016]. We recognized this and attempted to garner tweets while limiting the selection to those sent through Android, and found that it had been a while since Trump had used an Android to send a tweet. Trump's Twitter usage habits have changed since his inauguration as president, and the security parameters that he now has to follow may have included a switch to a different type of mobile device or a different platform entirely. Trump now appears to tweet mainly using Twitter for iPhone; a casual human analysis suggests that both Trump's personal bombastic authorship and the less hyperbolic promotional and 'official' staffers' tweets are being sent through iPhone in 2017. This means that the corpus of Trump tweets used (the maximum 3200 most recent tweets that the Twitter API allows a developer to scrape from an account) in the data may not have one author, but in fact, many [Roesslein, 2009]. This alone poses a serious problem for the analysis and identification of Trump parody, especially because some of the tweets that appear to use 'Trump voice' are off enough or contain enough promotional material that they may have just been issued by a staffer imitating the president's style.

The parody tweets, on the other hand, compose the majority of the classifier's test set and constitute various accounts attempting to use 'Trump voice' or otherwise parody Trump's Twitter style. These include:

- @DungeonsDonald [@DungeonsDonald]
- @realDonalDrumpf [@realDonalDrumpf]
- @RealDonaldTrFan [@RealDonaldTrFan]
- @realNFLTrump [@realNFLTrump]
- @Writeintrump [@Writeintrump]

Some of these are parodies in Trump voice explicitly about politics, and make frequent reference to entities and events that Trump himself tweets about or that surface in the media discourse surrounding Trump. Two, @DungeonsDonald and @realNFLTrump, use other content while retaining the 'Trump voice', and thus make humorous parodies of Trump that ask the reader to suspend disbelief and imagine Trump either engaging seriously with tabletop games or engaging seriously with football. [3]

These, too, however, have fundamental issues which might pose an issue for authorial analysis methods on parody identification. Some of these parodies are what a casual human analysis might describe as 'not good', indicating a failed parody that isn't necessarily immediately recognizable for using 'Trump voice'. @realDonalDrumpf, for example, rarely utilizes 'Trump voice' and prefers instead to rag on aspects of Trump's policy and appearance in a critical tone; @RealDonaldTrFan misuses 'Trump voice' and writes clusters of words in all-caps throughout the tweet as opposed to Trump's notable First Letter Capitalization of noun phrases interspersed with one or two all-caps phrases per tweet. Whether or not a parody of an author is 'good' parody is, of course, a subjective judgment, but some accounts certainly do it better than others, irrespective of content. The inherent fluctuation between parody authors between what constitutes their parodic styles adds another dimension to the complexity of the task, and behooves us to examine each of the parody accounts in question individually.

[3] Beyond his well-known feud with quarterback Colin Kaepernick.

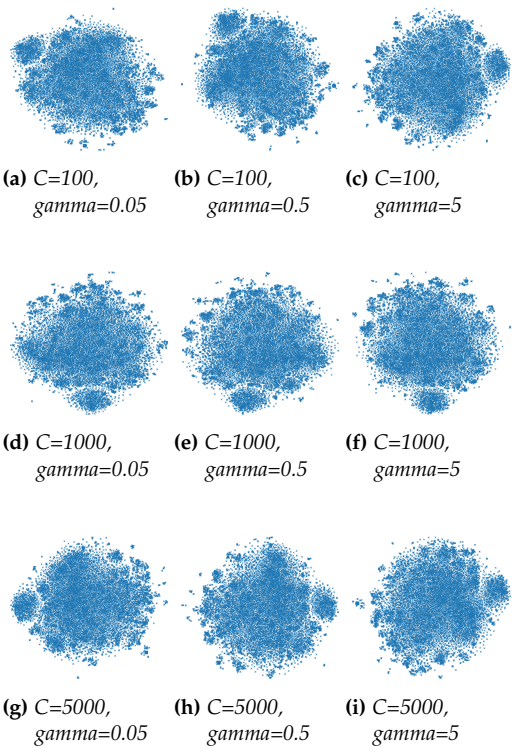
IV. Methods

We designed a pipeline resembling one that would be used for standard support vector machine authorship identification in order to construct classifier models from Trump and the other Twitter pundits and to test them against parody data. This pipeline does the following:

- Cross-validates the training data by adding several hundred randomly chosen tweets from the training authors to each test assessment of the model
- Tokenizes the sentences using NLTK's built-in function for tokenizing tweets, but being sure to retain handles, idiosyncracies of length and repetition, and the usage of irregular capitalization or punctuation [Bird, Loper, and Klein, 2009]
- Constructs a feature space from certain aspects of the tokens. Feature spaces included
  - word n-grams
  - part-of-speech n-grams
  - x-tag frequencies
  - frequencies of distance from dependency to head
  - frequencies of types of dependency relations (the last three using Carnegie Mellon's Ark Parser, which includes the TweeboParser dependency parser for tweets [Kong, Schneider, et al., 2014])
- Vectorizes the data using the feature space and the NLTK term frequency - inverse document frequency (TF-IDF) vectorizing function
- Trains the classifier on the tf-idf-vectorized data, with associated kernel function and kernel parameter settings
- Expands the feature matrix from a dense representation to a regular sparse one if possible to do within memory, and decomposes the matrix to 50 dimensions using NLTK's truncated singular value decomposition function if the sparse matrix cannot be held within memory
- Maps the sparse or decomposed matrix down to 2 dimensions using t-distributed stochastic neighbor embedding (t-SNE) and plots it
- Prints the classifier's evaluation of which test tweets belong to which author, organized by actual author and dividing responses into correct identifications of a tweet (the tweet is correctly identified by the classifier as belonging to author x, or the tweet is by an account parodying author x) and incorrect identifications (the tweet is incorrectly identified by the classifier as belonging to author y), while retaining the breakdown of wrong guesses by guessed author because this data is interesting

Much of the work that went into editing and trimming the model occurred in the application of different features and especially the use of different parameters for the classifier kernel. Messing with the kernel parameters helped to illustrate at first the deep problems with the data - maintaining a constant feature space but applying different C and gamma parameters showed that some combination of 'not good' Trump parodies and the perhaps multiple authorial (often similar, as staffers are possibly intentionally 'Trumping' their messages when tweeting from Trump's account) styles that are actually tweeted from @realDonaldTrump forces the identification of certain parody tweets under certain parametric conditions to non-Trump authors. Figure 1 depicts the t-SNE 2-dimensionalization of tweets transformed to the feature space defined by trigrams of NLTK's part-of-speech tagger. Table 1 offers a specific view of tweets by @DungeonsDonald correctly identified as parodying Trump versus those incorrectly identified as Mike Huckabee or Ross Douthat, who exert stronger or weaker pulls on the data as the parameters change. The effects on the test data are especially pronounced, which we seeded with cross-validating examples actually taken from the authors in the training set that we expected to match near-perfectly to their authors. Instead, some parameter configurations thoroughly fail to reliably assign tweets in the

test data to their respective authors - for example, when C=100 and gamma=5, Jonathan Chait's tweets are overwhelmingly (302 out of 500) assigned to Ross Douthat, whereas only 27 of 500 are assigned to Chait.



(a) *C=100, gamma=0.05* **(b)** *C=100, gamma=0.5* **(c)** *C=100, gamma=5*

(d) *C=1000, gamma=0.05* **(e)** *C=1000, gamma=0.5* **(f)** *C=1000, gamma=5*

(g) *C=5000, gamma=0.05* **(h)** *C=5000, gamma=0.5* **(i)** *C=5000, gamma=5*

Figure 1: *Part-of-speech trigram t-SNE visualizations of the test set under different parameters*

V. Results

The results of the experiment themselves were afflicted by the pulls that the training authors had on each other throughout the testing of various feature spaces. None of the classifiers were able to reliably distinguish Trump parody tweets from non-parody tweets, none identified the seeded training tweets in the test sets as belonging to their respective authors with reliability, and none ever received an NLTK-calculated classifier score above around 77 percent - and numbers that high only showed up in cases where the parameters were such that

Table 1: *Table 1: (Trump, Huckabee, Douthat) percentage point triples for @DungeonsDonald tweets trained on part-of-speech trigrams*

	g=0.05	g=0.5	g=.5
C=100	28.3, 27.9, 15.7	27.3, 27.5, 16.9	44.2, 6.74, 44.8
C=1000	25.7, 29.6, 12.2	27.3, 27.5, 14.8	45.3, 7.67, 43.1
C=5000	25.3, 26.8, 14.0	26.9, 27.2, 16.9	44.9, 7.12, 44.2

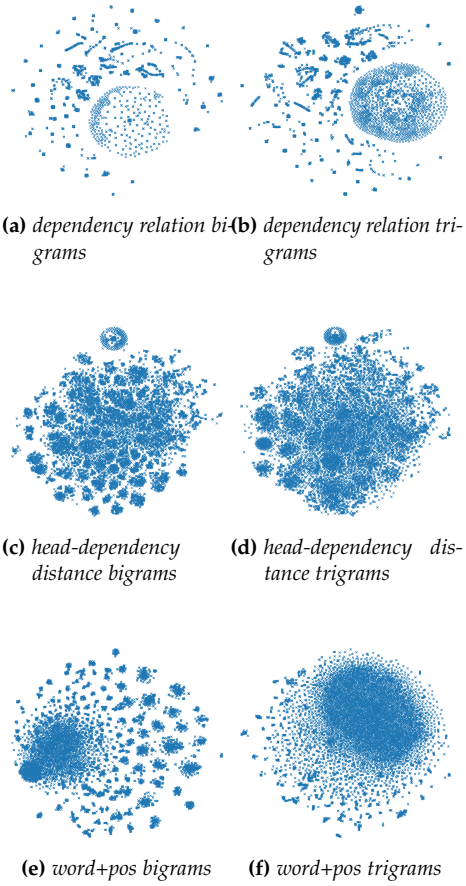
every tweet in the test set was guessed to be a parody of Trump. Figure 2 displays some of the t-SNE dimensionalizations of the classification of the test set with various feature spaces, and Table 2 illustrates the results of one of the best feature spaces found (noteworthy because each parody account actually does manage to have a plurality of their tweets guessed to be Trump, which was rare for all five accounts in most feature spaces). Using a feature space that was sensitive to syntactic concerns like the distance of the dependency to the head of each token and the frequency with which dependency relations like that of a subordinating conjunction to the head of the following clause made no significant or positive difference in the results when compared against style-agnostic featural spaces; the only significant variations between content-only and style-sensitive classifiers with similar or the same classifier parameters appeared to emerge because adding features and thus dimensions to the space while keeping the parameters the same necessarily implies that the kernel function will produce different classifications. As such, without any evidence to indicate a positive difference between the ability of a style-sensitive feature space to perform parody identification over a style-insensitive one, we must assume the null hypothesis.

VI. Conclusion

The primary reason for the failure of this experiment to produce useful results about the applicability of authorship attribution to parody identification was the data. Issues with the data stemmed from several sources. The

Table 2: *Table 2: Percentages of parody tweets correctly identified as Trump vs percentages of parody tweets incorrectly identified under a feature space including word and part-of-speech bigrams*

	Trump	Douthat	Chait	Yglesias	Huckabee	Ioffe
@DungeonsDonald	45.7	19.5	11.6	11.8	9.93	1.49
@realDonalDrumpf	30.3	24.2	12.2	9.39	17.2	6.66
@RealDonaldTr	27.4	17.9	7.95	21.1	21.3	4.25
@realNFLTrump	34.9	19.4	11.1	15.1	17.5	1.98
@Writeintrump	29.7	27.2	10.5	9.24	19.5	3.93



(a) *dependency relation bigrams* **(b)** *dependency relation trigrams*

(c) *head-dependency distance bigrams* **(d)** *head-dependency distance trigrams*

(e) *word+pos bigrams* **(f)** *word+pos trigrams*

Figure 2: *t-SNE visualizations of classifications using various feature spaces*

presence of other authorial voices mixed into the Trump tweets probably significantly threw off the ability of the classifier to build a consistent profile using any type of feature space, but especially those related to stylistic choices and linguistic idiosyncracies - exactly what the experiment attempted to isolate in the first place. Moreover, the inclusion of 'bad' parodies in the dataset meant that a handful of accounts repeatedly had their tweets classified as belonging to another author; for example, @realDonalDrumpf's guessed (incorrectly) score for Julia Ioffe was significantly higher than its guessed (correctly) score for Trump in feature spaces across the board; the propensity of @DungeonsDonald to be identified as Mike Huckabee also extended with varying severity depending on the specific parameters and features to @RealDonaldTrFan and @Writeintrump. To some extent, failures to correctly identify @DungeonsDonald (and @realNFLTrump) tweets might be ascribed to the feature space not being style-sensitive enough - these content-dissimilar accounts were banking on their use of 'Trump voice' being recognized, and to their credit, they are fairly good at replicating it (@DungeonsDonald makes a point to end sentences with infinitives: "The D&D Healing system is imploding... Players should agree to my rules to fix!", while @realNFLTrump dutifully replicates Trump's quirk of tweet-final one-or-two word phrases, e.g. "Sad!" or "Not good!"). However, the inclusion of multiple voices under one 'Trump' classification and the failure of some of the parodies to write in 'Trump voice' should be considered leading causes for the way the results turned out - any improvement on the experiment will have to begin there, and the experimenters will have to find a corpus by an author that uses a single voice that has

generated as many parodies for testing as has Trump.

Other aspects of the experiment, in addition to the fundamental issues with the data, hampered its progress as well. Our fairly simple feature spaces attempted to contrast style-sensitive approaches towards parody identification with style-insensitive approaches. Future instances of the experiment should consider looking further afield for features to extract, and may want to take into consideration some of the quirkier writing habits of an author before considering what means could extract such features from the text. Experimenters looking to replicate and improve upon this work should carefully consider what parts of a text can be considered style and what can be considered content, and how to construct and evaluate meaningful comparisons between the two.

Finally, the lack of any research grounding or prefiguring the application of authorial attribution techniques to parody identification was something that we attempted, with little success, to rectify. Fortunately, the paucity of prior research means that many different avenues towards solving this problem can be conceptualized, and that many different methods can be created and applied once sufficient sets of authors and parodies are found.

Acknowledgments

I would like to thank Professor Drago Radev for all his help in making this project succeed.

References

[Tschuggnall and Specht, 2014] Tschuggnall, M. and Specht, G. (2014). Enhancing authorship attribution by using syntax tree profiles. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2:195–199.

[Gamon, 2004] Gamon, M. (2004). Linguistic correlates of style: authorship classification with deep linguistic analysis features. *Proceedings of Coling 2004*, 611–617.

[Layton, Watters, and Dazeley, 2010] Layton, R., Watters, P., and Dazeley, R. (2010). Authorship Attribution for Twitter in 140 characters or less. *Proceedings of the 2010 Second Cybercrime and Trustworthy Computing Workshop*, 1–8.

[Raghavan, Kovashka, and Mooney, 2010] Raghavan, S., Kovasha, A., and Mooney, R. (2010). Authorship Attribution using Probabilistic Context-free Grammars. *Proceedings of the ACL 2010 Conference Short Papers*, 38 – 42.

[Uzuner and Katz, 2005] Uzuner, Ö. and Katz, B. (2005). A Comparative Study of Language Models for Book and Author Recognition. *Proceedings of the Second International Joint Conference on Natural Language Processing*, 969 – 980.

[Robinson, 2016] Robinson, D. (2016). Text analysis of Trump's tweets confirms he only writes the (angrier) Android half. *Variance Explained*, August 9, 2016, http://varianceexplained.org/r/trump-tweets/.

[Roesslein, 2009] Roesslein, J. (2009). Tweepy: An easy-to-use Python library for accessing the Twitter API. Github repository at http://github.com/tweepy/tweepy.

[Pedregosa, F., Varoquaux, G., et al., 2011] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825 – 2830

[Bird, Loper, and Klein, 2009] Bird, S., Loper, E., and Klein, E. (2009). Natural Language Processing Toolkit. O'Reilly Media Inc. http://nltk.org/book

[Kong, Schneider, et al., 2014] Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., and Smith, A. (2014). A Dependency Parser for Tweets *Proceedings of EMNLP 2014*,

[Trump, 2017] Tweets from Trump, Donald J., as @realDonaldTrump (Feb. 2017 – Oct. 2017). http://twitter.com/realDonaldTrump

[Ioffe, 2017] Tweets from Ioffe, Julia, as @juliaioffe (Jan. 2017 – Oct. 2017). http://twitter.com/juliaioffe

[Chait, 2017] Tweets from Chait, Jonathan, as @jonathanchait (Apr. 2017 – Oct. 2017) http://twitter.com/jonathanchait

[Douthat, 2017] Tweets from Douthat, Ross, as @DouthatNYT (Mar. 2017 – Oct. 2017) http://twitter.com/DouthatNYT

[Huckabee, 2017] Tweets from Huckabee, Mike, as @GovMikeHuckabee (2015 – 2017) http://twitter.com/GovMikeHuckabee

[Yglesias, 2017] Tweets from Yglesias, Matt, as @mattyglesias (Jun. 2017 – Oct. 2017) http://twitter.com/mattyglesias

[@DungeonsDonald] Tweets from an unverified user posting as @DungeonsDonald (May 2016 – Oct. 2017) http://twitter.com/DungeonsDonald

[@realDonalDrumpf] Tweets from an unverified user posting as @realDonalDrumpf (Feb. 2017 – Oct. 2017) http://twitter.com/realDonalDrumpf

[@RealDonaldTrFan] Tweets from an unverified user posting as @RealDonaldTrFan (Jan. 2017 – Oct. 2017) http://twitter.com/RealDonaldTrFan

[@Writeintrump] Tweets from an unverified user posting as @Writeintrump (Jan. 2016 – Oct. 2017) http://twitter.com/Writeintrump

[@realNFLTrump] Tweets from an unverified user posting as @realNFLTrump (Jan. 2016 – Oct. 2017) http://twitter.com/realNFLTrump