Automatic Lie Detection in Political Speech

Mauricio Wulfovich
Dept. of Computer Science
Stanford University
mauriw@stanford.edu

Nicholas Barbier
Dept. of Computer Science
Stanford University
nbarb@stanford.edu

Grant Russell
Dept. of Computer Science
Stanford University
grantrus@stanford.edu

Abstract

Due to the ease of information dissemination on social media, fake news and deceptive political information have become increasingly problematic. Deception detection provides obstacles for automation due to its high degree of domain specificity and semantic interpretation, as opposed to other natural language understanding (NLU) tasks that rely on analysis of relatively more local phenomena.

Existing models have attempted to recognize deceptive speech through a variety of independent approaches, including linear analysis on counts of appearance of certain word-categories, bag of words regression on large annotated datasets, and automatic fact-checking by cross-referencing on external domains. Our strategy develops an ensemble neural model that trains on a concatenated statement representation combining contextual word embeddings, meta-data encodings, and feature representations drawn from the linguistic literature – a slew of the best extant approaches.

1 Introduction

Work on lying spans several academic domains and research areas, from theory of mind (a lie presumes an appearance-reality dichotomy), to affective psychology (emotional and behavioral markers of lying), to sociology and economics (exploring justifications for cooperation and non-cooperation among communal social agents).

Of particular interest are the linguistic markers of lying. Newman et al. (2003) begin their analytics forays with an exploration of the socio-linguistic theories about the production of reality vs. memory of reality (reference of speech, truthful or deceitful), as well as theories about the affective and cognitive load of the act itself.

Analyses of the markers of lying can lead to lie-detection algorithms and models that predict

Figure 1: Count of number of singular first person pronouns in deflecting, untruthful statements by Anthony Wiener (right), and in a press conference confession of guilt (left) https://www.youtube.com/watch?v=Vc073RIC7_M

or classify the veracity of a statement automatically. Although automatic lie-detection has utility is many different contexts (think poker, marriage, academic research), one salient domain of interest is that of political lie-detection. Causes include the larger scope and repercussions of misrepresentation of the facts and the frequent use of deceptive speech in order to curry rhetorical favor.

Furthermore, the last Presidential election cycle in the United States saw a disinformation campaign and the inception of mass distribution of ”fake-news” via online channels, spurring interest in the development of automatic filters for fake-news detection, a problem generally related to that of lie-detection.

2 Previous Work

Prior research in the field of lie detection falls into two classes; qualitative (linguistic) and quantitative (computational modelling). First we will address the qualitative literature spanning from the 1990s into the 2000s, and second we will address the quantitative literature spanning the late 2000s into the early 2010s.

An early formal exposition on the markers of lying in Adams (1996) describes techniques to iden-

tify lying in written text prior to a witness interrogation, including include usage of possessive pronouns, nouns, verbs, extraneous information and the noted conviction of the speaker. The paper has the explicit purpose of identifying markers of lying in order to extract a confession from the witness.

The heuristics described in this work are directly cited by Newman et al. (2003) as an inspiration for the categories for comparison in the quantitative analysis that follows. Newman (2003) identifies several categories of interest, asks undergraduate research subjects to lie, and then performs linear regression on the counts of the linguistic categories of interest using the LIWC (Pennebaker et al., 2015). Likewise, Newman et al. (2003) is later cited in Mihalcea et. al (2009) who expands on the previous work by exploring betrayal in online communities.

The quantitative results are consistent with the domain expert hypotheses about lying and deception. In particular, the dissociation hypothesis explored in Newman et al. (2003), namely, that liars should tend to avoid first person language in order to extricate themselves from the guilt and negative emotion at large, as well as reflect the psychological distance to an imagined happening (as opposed to a real one), was proposed by Adams (1996), and quantitatively supported in Newman et al. (2003) and Mihalcea et al. (2009).

More specifically, Newman et al. (2003) discovered three main categorie of significance: limited usage of first person (and other) pronouns, negative-emotion words, and limited vocabulary indicative of cognitive complexity (”but”, ”except”, ”without”). Differing degrees of emotional involvement – research was performed with lies with low stakes in a laboratory setting – could account for limited generalization of the findings.

Finally, Newman et al.(2003) mention that these linguistic techniques may be used more specifically when comparing against a baseline, i.e. we can use them to note changes over time; a police investigator could use context built up over exposure from multiple interviews, writings, videotapes, etc. to try to learn more the speaker’s innate ”truth” style. Likely such techniques are less useful in analyzing temporally local speech.

Along with the qualitative research that identifies specific and discrete feature categories of speech for examination, a variety of computational

General prediction equation:			
All 5 studies combined		59/62/61**	8
First-person pronouns	.260		
Third-person pronouns	.250		
Negative emotion	–.217		
Exclusive words	.419		
Motion verbs	–.259		

Figure 2: Correlation with truth-telling (Pennebaker et al. 2015)

and deep learning models have emerged for lie-detection. With respect to machine learning models used to discern lying, Chow et al. (2017), Desai et al. (2017), and Wang (2017), all failed to train RNNs that significantly outperformed the baseline models. In fact, when used, bag-of-words SVMs and Naive Bayes outperformed the RNNs. And despite using various state of the art preprocessing methods and RNN models, neither Chow et al. (2017) nor Desai et al. (2017) were able to achieve over 62% accuracy, which is only 2% better than simply choosing truth for every label on the test set (it is skewed 60:40). Wang trained on the LIAR corpus, but his model, likewise only achieved a 2% better accuracy than choosing the majority label.

The authors RNNs were able to overfit to the data, showing that the model was properly capable of learning, however neither the CSC nor LIAR corpus contained enough data for the RNN to generalize. In Chow et al. (2017), despite reducing overfitting to increase the RNNs generalizability, their model still failed to generalize better than the aforementioned 62% on validation accuracy. This further indicates that there is an overall pattern of a data bottleneck with regards to training these sequencing models compared to bag-of-words models. This conclusion is affirmed in the paper by Oshikawa et al. (2018), where the more recent large scale datasets, such as the FEVER corpus, are mentioned as important steps in advancing the state of automated fake news detection.

3 Data

3.1 LIAR

We used the LIAR corpus developed by Wang (2017). It is a decade-long dataset of manually-labeled short political statements from the website Politifact.com. LIAR corpus provides a variety of labelled metadata (speaker name, state (physical location), venue, political party, subject,and more (11 total)) and truth judgements with respect to 12.8K statements rated over the past decade.

Statement: *The last quarter, it was just announced, our gross domestic product was below zero. Who ever heard of this? Its never below zero.*
Label: Pants on Fire
Speaker: Donald Trump
Speaker's job: President-Elect
State: New York
Venue: presidential announcement speech
Party: Republican
Subjects: economy, jobs

Statement: *"EPA officials have commended (Koch Industries) for our commitment to a cleaner environment and called us a model for other companies."*
Label: Barely True
Speaker: Charles Koch
Speaker's Job: Koch Industries
State: Kansas
Venue: op-ed for the "Wall Street Journal."
Party: None
Subjects: environment, market-regulation

Table 1: Random excerpts from LIAR dataset

Although the political statements themselves are abbreviated and contain partial quotes interleaved with third person commentary, the corpus also contains links to complete source documents of speeches.

The strength of the LIAR dataset specifically is the richness of its labelled data and metadata. Furthermore, the truth value of every statement has been evaluated by Politifact a priori, and eliminated an inherent difficult in deception-detection – it isnt clear how Mechanical Turkers or other dataset labelers could naively label a statement for its truth value since truth, in any case other than that of the most trivial facts and mathematical truisms is dependent on an understanding of complex and contextual domain specific knowledge.

3.2 LIWC

We also used the LIWC Corpus (Pennebaker et al., 2015) in order to get feature counts for a variety of relevant features. The LIWC corpus, mentioned in "Lying Words" (2003), and developed by co-author Pennebaker, provides a variety of linguistic categories, tokenization tools, and word lists for the parsing and linguistic feature annotation of text. Applied to the LIAR corpus, it yields an additional 93 feature categories, ranging from the previously mentioned "personal pronoun", to swears, religious words, punctuation count, and more.

Dataset Statistics	
Training set	10,269
Validation set size	1,284
Testing set size	1,283
Avg. statement length (tokens)	17.9
Top-3 Speaker Affiliations	
Democrats	4,150
Republicans	5,687
None (e.g., FB posts)	2,185

Table 2: The LIAR dataset statistics

4 Approach

4.1 Metrics

The problem of training is framed as a straightforward classification problem. Give a set of outputs – the labelled truth values of the Politifact data set, either normalized to 2,3, or 6 total values, provide a model that most accurately predicts the output given a new statement example with meta-data. Since we proceed using a classification framework, we report accuracy and F1 scores as metrics, although accuracy specifically will provide the strongest basis of comparison with existing results.

4.2 Statement-Level Features

A major component of our project was creating 18 hand-written linguistic feature functions. These were inspired by the categories in (Newman et al., 2003) as well as (Mihalcea and Strapparava, 2009) (Adams, 1996). The difficulty of complete tokenization, as well as creating a sufficiently complete dictionary for each feature category led us to the use of the LIWC 2015 corpus. The 93 feature categories on each label previous mentioned were computed at the statement level on the whole dataset. However, during training only a subset were used. First we used the subset of four previuosly mentioned in Newman et al. (2003), that showed the highest results for lie-detection with linear regression. Second, we performed classification using the full feature list (93 features). And finally, we performed classification using random feature lists of length 10 as baselines.

Further research would likely examine each of

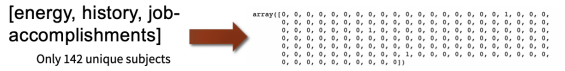

Figure 3: Example of disparate categorical data transformed to sparse encodings

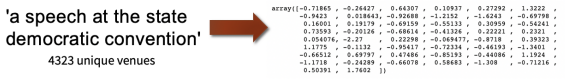

Figure 4: Tf-idf weighted sentence-average GLOVE embeddings

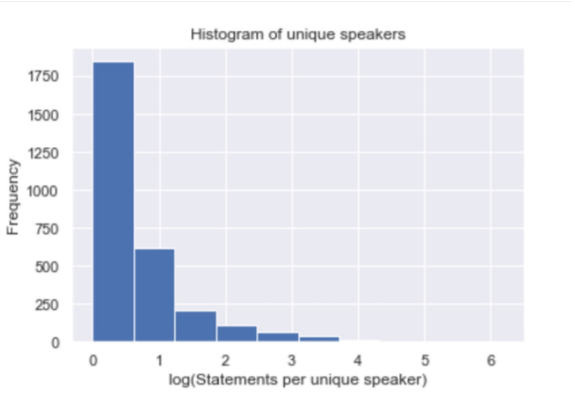

Figure 5: Meta-data distribution

the 93 categories without a prior qualitative belief about their strength and test them in conjunction with the more powerful models to follow; however, given the size of such a set which has $X = \{Features\}, |X| = 93, |\wp(X)| = 2^{|X|} = 2^{93} \approx 10^{27}$ possible subsets, using qualitative heuristics, is necessary. For examples of some of our linguistic feature functions, see table 3.

4.3 BERT

We extracted BERT contextual embeddings from the statement texts for the purposes of training. Pre-training the BERT embeddings took approximately 20 minutes for the training set (and proportional amounts of time for the other sets, i.e. 2 minutes each for validation and testing sets each) on a 2018 Macbook Pro with 32GB of RAM and a 6-Core Intel i9 Processor. We used Github user hanxiao's bert-as-a-service, and pre-batched the elements. Embeddings were returned with length of the sequence as the primary dimension on a per-statement level; means were taken across this dimension for purposes of tractability. Embedding files for the training data without dimensionality reduction were 33GB. Likely due to null suppression encoding on large padded blocks of 0, all the dimensionality reduced representations are on the order of hundreds of megabytes.

4.4 Meta-data encodings

For every statement in the dataset, there are 11 meta-labels. These include information on the statement subjects, speaker, speaker's job, state, political party, venue, and amount of lies the speaker's told in the past. The labels are heterogenous: 5 are numerical, and 6 textual. A major task in this project was discovering the best way to encode all of these forms of information in such a way to best inform our models.

After analysis, we discovered that there are roughly 2 types of distributions of the meta-data text. The first distribution corresponds to meta-data such as venue and speaker's job. The amount of unique labels is proportional to the amount of statements. For instance, there were 4342 unique venues in the 10,269 training examples. Furthermore in this distribution, the text roughly follows a zipf distribution. For example, 'a news release' is the venue for 248 of the training statements, yet only 5.5% of the venues occurred more than 5 times. Figure 5 shows this further.

The second distribution corresponds to disparate, sparse meta-data labels such as state, subject, and party. For example, there are only 142 categories of statements, 90% of which occur in more than 25 statements. These meta labels are also disparate– such as "civil-rights" and "families".

We built and tested two forms of metadata encodings to correspond to each of the distributions. For the meta-data following the second distribution, we encoded it as multi-label binarization (figure 3), due to its aforementioned disparate, high-overlap, nature. For every form of meta-data following the first distribution, we took the GLOVE embeddings of its words, tf-idf weighted them, and averaged the results over the sentence. Our hypthesis is that GLOVE embeddings generalize much better for all the unique, low-frequency meta-data. We used tf-idf weighted, sentence-average GLOVE embeddings, instead of the more complex BERT embeddings for a variety of reasons. One, most of our statements are noun-phrases where order is intuitively of less importance (e.g. 'florida town hall' vs 'town

Linguistic Feature Function	Examples
First person singular pronouns	me, my, we
Words >6 Letters	-
Impersonal Pronouns	it, it's, those
Articles	a, an, the
Confidence	trust, choice
Emotional Tone	bright, dark

Table 3: Example Linguistic Feature Functions

hall florida'). Secondly, previous research (Arora et al., 2016) has shown that such tf-idf averaged embeddings are a tough to beat sentence baseline. Lastly, obtaining such embeddings took on the order of seconds for the whole dataset, which is roughly 100X the speed of obtaining our BERT embeddings. Our results support our hypothesis: when testing our model with meta-data, choosing its respective encoding method outperformed using the other encoding method.

4.5 Label Simplification

Politifact ranks statements truth value on a 6-category scale. While such divisions create nuance (e.g. the different between 'pants on fire' and 'false'), they might cause issues. The linguistic and metadata differences between such fine categories might not be clear. The linguistic differences between roughly-true and roughly-false statements, however, might be more distinct, and a classifier might be able to distinguish them better.

We decided to test the above hypotheses by combining labels. We tested all of our models both on the original 6 class labels as well as with 3-class labels that were created by combining "pants-fire" with "false"; barely-true" with "half-true"; and "mostly-true" with "True".

4.6 Linear Models

We built two softmax models baselines by mapping each of the Politifact labels to a numerical values (ex: {0, 1,...,6} $\equiv$ {HALF-TRUE, false, mostly-true...}). One model is a bag of words model based on the statement text, and then another is a linear model based on the confidence of the bag of words model along with metadata and featurized elements of the statement itself. The features include cognitive complexity (on a custom list of complexity words), possessive pronouns, negation words (each measured by counts), and negative sentiment (using the nltk sentiment corpus) on a statement-by-statement basis. These specific features were chosen due to their prominence in Newman et al. as four categories of note in the linguistic literature related to (non-statistical) lie detection as well as in simple linear models with accuracy baselines. LIWC corpus was used in neural models (MLP) only as described in the next section.

4.7 Ensemble Models

Random Forest and Extra Trees are among two ensemble learning methods for classification that we utilized, which are constructed from a multitude of decision trees. The output of these models is the mode of the decision of the individual trees. These ensemble models have proven to be especially useful when only limited training data is available, where they typically out-perform other state-of-the-art models. As we face a similar problem in our lack of training data, these models were a clear solution. We used bag-of-word embeddings as input to both models, constraining our trees with a maximum depth of 200 and n_estimator values of 800. The n_estimator values represent the number of individual trees in the forest.

4.8 Neural Models

The simplest neural models were multi-layer perceptrons with a single hidden layer and hidden dimensionality on the order of the per-sentence averaged BERT-embeddings (500). Due to the high performance of these models and the fact that the addition of LIWC corpus feature information decreased the test set performance (these results will be mentioned at greater depth in the following section review of the training results), we eliminated the use of feature function in the more complex neural models.

In the development of the higher complexity neural models (RNN and CNN architecture) we had a few main goals. First was increases classification accuracy compared to the relative strength of the neural baseline found in MLP. Second (relatedly) was preventing overfitting and introduction of excessive complexity in general through multiple hidden layers, activation functions, etc.

Wang (2017) provides an excellent neural baseline for usage with the LIAR dataset as described in his paper. This model concatenates a CNN and Bi-LSTM on metadata with a max-pooled CNN enriched by word embeddings on the statement

Models	3-Class	6-Class
Majority Label	.370	.208
Softmax	.403	.247
Random Forest	.441	**.275**
Extra Trees	**.450**	.271
RNN	.430	.265
BERT RNN	.381	.234
BERT MLP	.439	.263
BERT* Temporal CNN	.371	.208
Softmax + Metadata + Ling. Features	.439	.277
BERT MLP + Metadata	**.472**	**.292**
BERT MLP + Metadata + Ling. Features	.361	.160

Table 4: F1 score evaluation results on test data. Top section: text-only models. Middle section: text-only neural models with BERT embeddings. Bottom: ensemble models.

text. Our hypothesis holds that the enriched meta-data and BERT representations together can improve on the neural model baseline as given. Our appraoch follows a commonly used pattern for incremental improvement in neural models, by using supplementation with feature specific information relevant to the task – in this case, features informed by meta-data representations and contextual embeddings on the words.

The created networks utilize long short-term memory (LSTM) cells, which are based on RNN architecture. The first model pre-processes the statements using the bag of words method to turn the statements into sparse vector representations. We then pass this into a keras embedding layer, which will shape the input to be properly passed in to the LSTM cell. We then specify that we want the LSTM cell to keep its output time sequenced, as we will be passing it into another LSTM cell after connecting it through a dense, or fully connected layer. Having multiple LSTM cells adds complexity to the model in hopes that it will better comprehend the complex speech patterns being used. Next, we connect the 2nd LSTM layer to a 2nd fully connected layer, before connecting to a final dense layer with a dimensionality equal to the number of labels, representing the confidence of each. We run this through a softmax activation, which returns a probability distribution of each class. In order to pick our label, we choose the option with the highest probability.

The second RNN model utilizes this same exact structure, but instead of bag of words representations of statements, we are using pre-processed bert embeddings. These embeddings were then taken and used as input into the first layer of the LSTM model described. Early stopping was also employed, as val accuracy appeared to plateau fairly quickly and unpredictably in the LSTM models, perhaps due to inconsistent initialization.

5 Results & Discussion

5.1 Comparison

As expected with our limited training data, the Random Forest and Extra Trees models performed exceedingly well. Trained only on bag-of-words embeddings and with trees of 200 depth, Random Forest was able to each 27.5% accuracy, outperforming most other models

The more surprising results came from our T-CNN and BERT MLP + Metadata models, with T-CNN under-performing and the BERT MLP model over-performing over our expectations. Two T-CNN models were trained, both shallow and deep, and both achieved the same accuracy on the test set. This is because both models fundamentally failed to learn to generalize, simply picking the majority label ”3” for every input example. This explains why the performance exactly matches that of the Majority Label model. There can be many causes for this, but despite using normalization techniques, shallower models and many other methods, the accuracy failed to increase. Thus, we can argue that the likely cause was due to a lack of training data for the T-CNN model to learn from.

The BERT MLP model was the surprising over-

performer. We continued to train this MLP model until it greatly overfit the training data, achieving 40% accuracy on the training set. To our initial surprise, this aided the model in its ability to generalize, achieving 29% accuracy on the test set, but only 24% accuracy on the validation set. This could potentially be an outlier due to the limited sample size of our sets.

Other models such as RNN failed to perform as well as MLP and Random Forest. We expected RNN to exceed in this task of text sequence classification, as RNN works very well on temporally-structured data. However, seemingly due to similar reasons to the CNN failing, the model failed to generalize. Although there was variance in its answers, the vast majority of results were also the majority label "3", showing that it failed to learn the underlying structure of the data in a significant way. This leads us to believe the underlying reason for the model failing is also due to a lack of training data.

5.2 Sources of Error

Although deception-detection occurs at the statement level, as previously mentioned, the statement-level examples of the LIAR corpus provide direct quotes from the Political candidate interleaved with third party commentary. For example:

"Hillary Clinton agrees with John McCain 'by voting to give George Bush the benefit of the doubt on Iran.'"

Important for purposes of distillation, abbreviation, and contextualization, the third party commentary likely proves useful for a politically interest web surfer to quickly understand the content of some political statement, however, as an input into a classification task, it provides confounding error in two ways.

Firstly, from on the semantic level it contrasts the possible statement-level lie with a meta-statement-level certainty. For example, even if the quoted statement itself is untrue, the third person description of the assertion is certainly true (assuming a baseline of trust in Politifact). And secondly (relatedly), from a syntactic perspective it diffuses the linguistic content of the lie itself from the documental content of the commentary that contains it.

For example, even if personal pronoun usage were a surefire heuristic for lie detection, the pre-

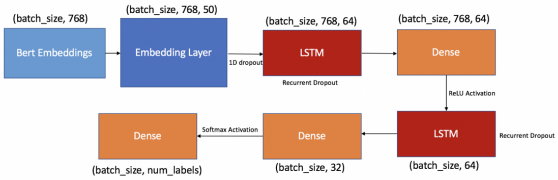

Figure 6: RNN Architecture

sentation of statements in the LIAR corpus drawn from Politifact would possibly miss such cues due to abbreviation of the full statement text.

6 Conclusion

Through investigations of predictive models for political deception using statement-level representations, word-level features, and metadata encodings, we developed several insights into the problem of lie-detection.

6.1 Label Simplification

Both 3-class and 6-class classification's best models created an approximately 10% increase in the test set compared to the choose-majority model. As such, our hypothesis that 3-class models would generalize better than 6-class models isn't correct. In the future, we'd like to do further testing on 2-class and 4-class models, as well as on the linguistic differences in finely grained truth categories.

6.2 Sentence Features, Meta-data, and Neural Models

First was the strength of BERT-only models. Using BERT with MLP and a single hidden layer we achieved score of 0.263, much higher than baseline and majority. However, as shown in the results table, the classification score decreased on BERT (sentence-average) embeddings when concatenating with the specific subset of LIWC of interest, and even more with a random subset of features, and performed worst of all with the full set of features. Our conclusions about these results informed the more complex models that followed. In particular, we hypothesize that the BERT embeddings already capture sufficiently complex and finely-grained representations of on the word and statement-level that is only redundantly, and more coarse-grainedly captured by the feature functions.

Although we originally planned to develop an ensemble model with a concatenation of statement

level representation (through BERT) with feature-functions or other statement level word category counts, along with metadata, we decided against this approach. Instead we combined the best of the statement and encoded meta-data information.

Furthermore, answering the original research question, it's clear that the high results for the concatenated neural models show promise for the development of automatic lie detection using deep-learning frameworks. In particular, we found that the higher the complexity of the statement-level representations (BERT) and the meta-data contextual representational encodings, the higher the performance of RNN and CNN architectures.

We hypothesize that developing more complete sets of relevant meta-data and statement text may raise accuracy even higher, without extending this work to become a semantic fact-checker as mentioned in the subsequent section. For example, extending the contextual embedding to a longer portion of the speech, and extending the meta-data with further relevant biographical data about the speaker (that could be queried, for example, from Wikipedia, or other public knowledge databases about public political speakers), would improve the model accuracy under this hypothesis.

6.3 Future Work & Limitations

Using a bag of words approach (even after embedding) to understand deception is text only provides a partial perspective, especially due to the dependent on larger context for the semantic content of a lie. We used a political dataset of interest in the current research; future work should extend to lying in other domains. It's not clear how such a rich and diverse dataset would be annotated – the relative importance of political speech motivated the development of Politifact, but in other domains, lying is either a more private, or lower stakes enterprise.

Mihalcea et al. (2009) describe a case-study that provides pre-annotated data, that of online games with full log records. An extension of such research to online communities at large (which, for example, in Reddit alone, cover a staggering breadth of topics) would provide both the diversity of examples, and the annotation – users usually do not tolerate deception online if they can detect it – necessary to extend this work.

Furthermore, as mentioned in the previous section data-augmentation in existing domains would help, as would knowledge graphs about the statements at play. Here we should make the distinction again between semantic and syntactic level lie detection. The latter detects lying by its linguistic epiphenomena, while the former directly evaluates truth by querying some implicit knowledge representations in large graphs. Because of the difficulties of evaluating symbolic queries (as show, for example, by the difficulty of non-local NLI queries), its unlikely that this work will yield fruitful results, at least not as an extension of the existing work shown in this paper.

7 Github Repository

https://github.com/grantrus/224ULieDetect

Acknowledgments

We would like to thank out project mentor, Ashkon Farhangi, as well as the course instructor, Chris Potts, for valuable guidance and feedback in the course of this project.

Authorship Statement

Mauricio: research, neural models, meta-data encoding, data pre-processing. Nicholas: research, paper writing, feature function LIWC, simple MLP, BERT embeddings. Grant: research, neural models, random algorithms, results analysis.

References

Susan H Adams. 1996. Statement analysis: What do suspects' words really reveal. *FBI L. Enforcement Bull.*, 65:12.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings.

A. Chow and J. Louie. 2017. Detecting lies via speech pattern. *Stanford CS224s Class Project*.

S. Desai, M. Siegelman, and Z. Maurer. 2017. Neural lie detection with the csc deceptive speech dataset. *Stanford CS224s Class Project*.

Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 309–312. Association for Computational Linguistics.

Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. 2003. Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665–675.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.

James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. The development and psychometric properties of liwc2015. Technical report.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.