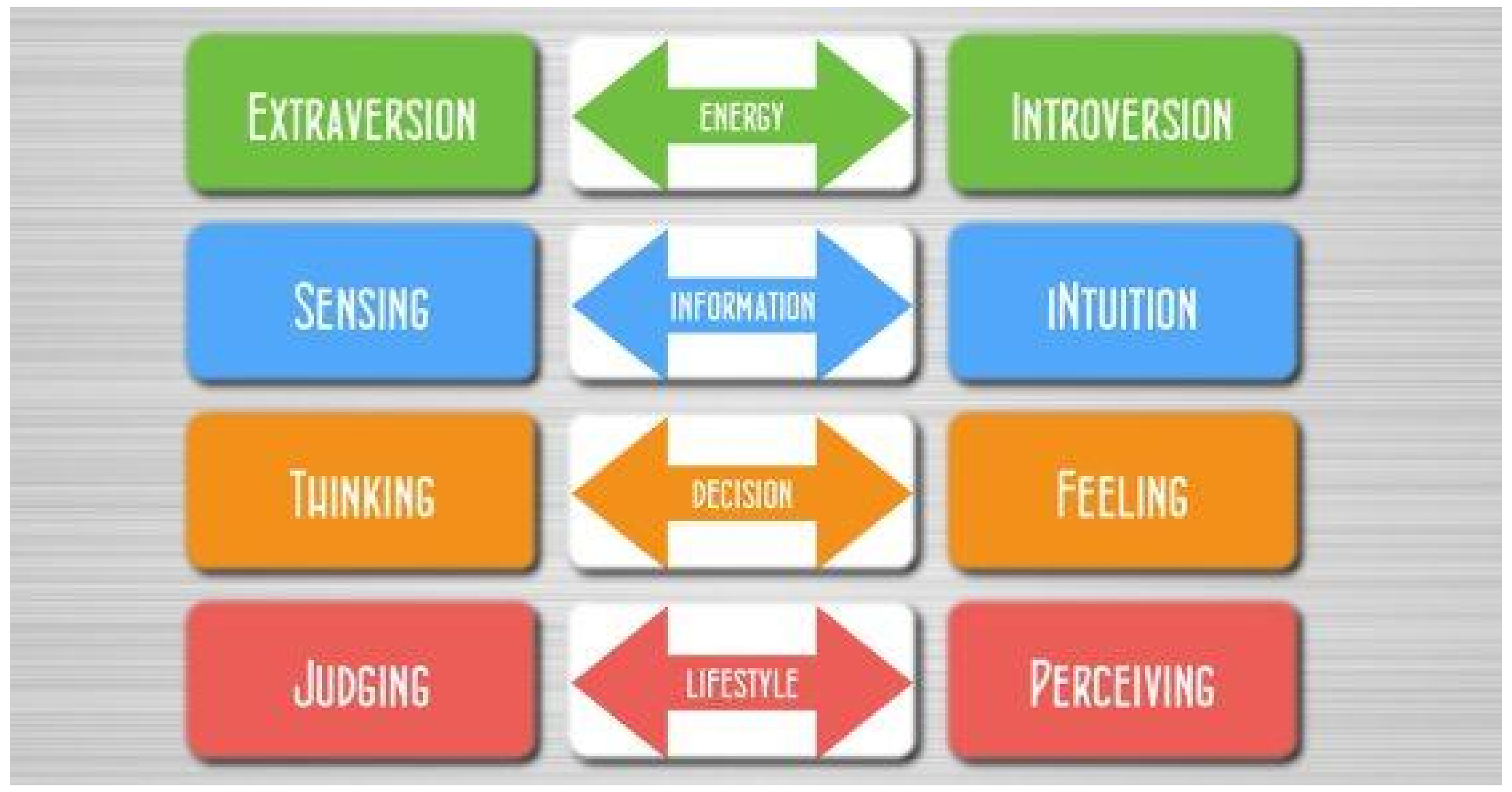
Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI Personality Type Prediction

Brandon Cui, Calvin Qi

Motivation

The Myers-Briggs Type Indicator (MBTI) is one of the most widely used descriptors of personality type. It describes the way people behave and interact with the world around them with four categories and 16 total types.

In a world where communication is increasingly social media based, we are interested in finding out if there is a strong relationship between one's use of language online and their actual personality.



Data

ENTP: "I'm finding the lack of me in these posts very alarming."

INFJ: "What? There's a series! Thanks for letting me know :)"

Dataset:
50 most recent posts of 8675 users from on PersonalityCafe forum. We partitioned all 422,845 posts to be classified with their MBTI personality type.

Preprocessing:
After analyzing the raw data and getting weak results on unprocessed versions, we added many steps to handle "internet lingo" by lemmatizing, standardizing punctuation, tagging URLs/emojis, etc.

Imbalances:
The data was strongly skewed in favor of certain personality types (90,000 ENFP vs 2,000 ESFJ). Data augmentation and removal were used to remedy this.

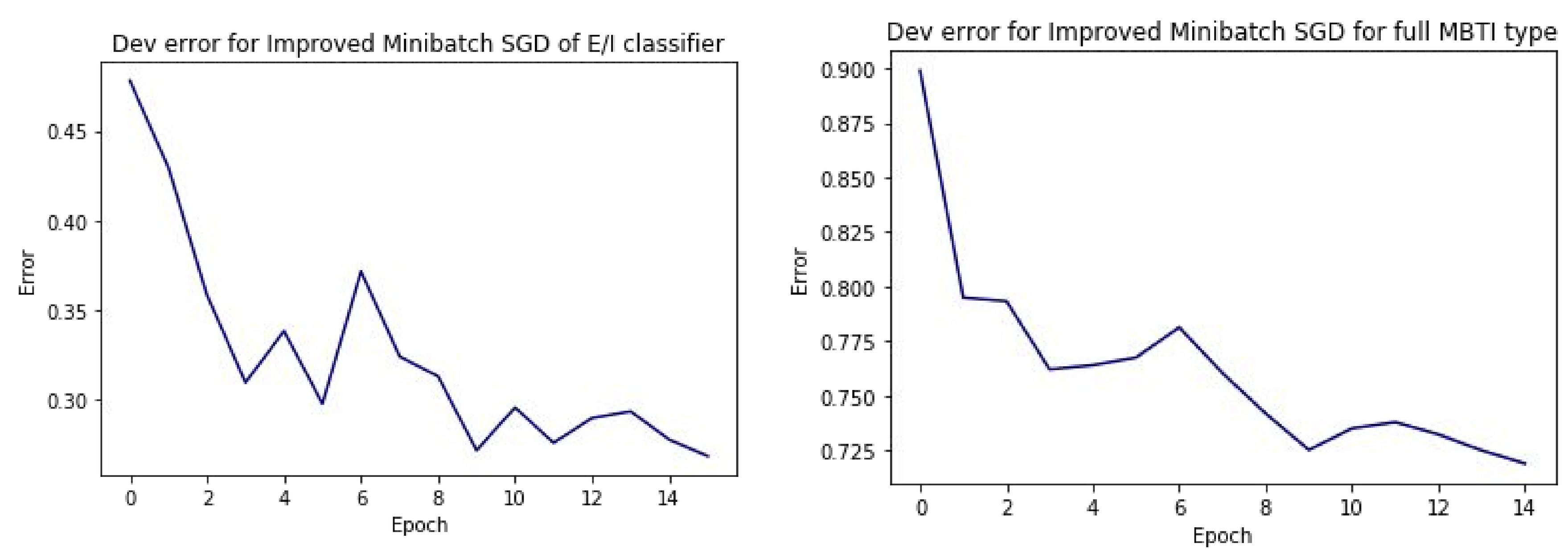
Methodology

Baseline:
Multiclass Softmax Classifier with basic bag of words model and no preprocessing: ***17% test accuracy = 83% error***

Improved Approaches (without deep learning):

- More sophisticated language preprocessing layers
- Balancing the training set with *augmentation+removal*
- Setting aside a portion of data for *hold-out* cross validation
- Training *four separate regularized SVM* binary classifiers with SGD and aggregating them to get overall MBTI result
- Tweaking regularization rate and minibatch size and bag-of-words size relative to word frequency
- *Additional features* from bigrams, skip-bigrams, part of speech tags, capitalization

Error Plots during training for a single personality category (left) and all four (right)

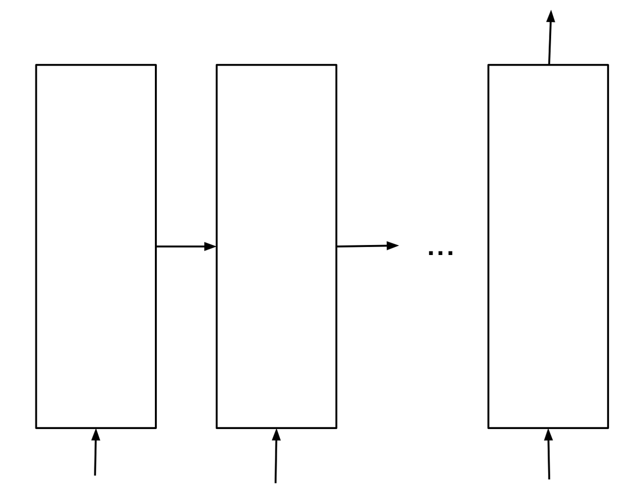


Results

Error:
Using the strategy of combining four category classifiers to predict the overall type: ***32.6% test accuracy, 78.4% error***

Classifier Type	Number of posts per label	Test Error
E/I	97582 / 325263	0.1827
S/N	58023 / 364822	0.1123
T/F	193533 / 229312	0.2913
J/P	167110 / 255735	0.3613
Overall	422845 total	0.3261

Deep Learning



Encoder: Long Short Term Memory (LSTM)
Used a LSTM to encode every phrase, actively learning embeddings.
Decoder: The decoder was a 3 layer neural network, with rectified linear units (ReLu) activation functions and a softmax layer on the last layer to obtain class probabilities.
Loss Function: we used cross-entropy loss for our loss function (1):

Hyperparameters: Dropout, Number of hidden encoding layers, hidden size, embedding size

16 Class Classifier
When classifying all 16 MBTI classes together, we ran into similar issues with traditional machine learning methods, where we were unable to well learn the dataset. Overall, we were able to achieve **38% test accuracy** => **62% test error** after random hyperparameter search (Bergstra et al. 2012).

Binary Classifier
Below we present some of the results our deep learning algorithm was able to achieve, the bolded results indicate the best performance (table 1):

Classifier Type	Embedding Size	Hidden Size	Dropout	# Hidden Encoding Layers	Dev Accuracy	Test Accuracy
E/I	256	256	0.1	1	**0.8974**	**0.8951**
E/I	128	300	0.15	1	0.8905	0.8892
S/N	256	256	0.1	1	**0.8856**	**0.89848**
S/N	200	300	0.15	1	0.8691	0.86656
T/F	512	256	0.1	1	0.6910	**0.6909**
T/F	256	256	0.15	1	**0.6912**	0.6848
J/P	256	256	0.15	1	**0.6605**	**0.6765**
J/P	128	300	0.1	1	0.6594	0.6837

Future Approaches
We plan on using pre-trained GloVe embeddings that are better able to capture temporal information along with K-char embeddings to try to gain better insight into the smaller intricacies of the dataset.

Analysis

- Deep learning outperforms conventional ML approaches by a few percent.
- Error analysis shows that data preprocessing is the most influential step
- Final accuracy of 38%, which is a significant improvement over our baseline but isn't nearly perfect
  - The problem itself has no clear oracle or human solution
  - Unlike vision or sentiment tasks, a human would not be able to accurately predict nuanced personality types by just looking at these posts
  - The machine is finding patterns that humans can't
- In fact, there may not be a strong connection between one's language use in an online persona and their actual non-virtual personality.

Future Work

- Currently trying different representations of language besides bag of words
- Incorporate information about the *user* and their total history of posts instead of treating each post independently
- Try unsupervised learning to see if people naturally cluster into personality types and compare those to the MBTI classes to see if they relate
- Find data that includes *context* of the conversation, e.g. surrounding posts
- And, of course, more sophisticated neural network architectures can help