Feature Engineering and Word Embedding Impacts for Automatic Personality Detection on Instant Message

Imanuel Buhapoda Drexel
Warung Pintar Sekali
Jakarta, Indonesia
imanuel@warungpintar.co

***Abstract*— Automatic personality recognition (APR) is a task to detect the personality of people using computerized method. Thanks to the advancement of technology these days, there are ample sources of data that could be collected to do this task especially ones with user generated data like instant messaging. We proposed the method for feature engineering impact on APR task done in the short messages type of data. The concatenation of the messages for each subject detected in the dataset has significant improvement result besides doing the preprocessing steps for each line of the text. This research enriches the current methodology and gives another option for data sources compared to previous study. It is also valuable for a company to profile its customer aiming in finding the best services for each person in his personality type.**

Keywords— feature engineering, big five personality, personality recognition

I. INTRODUCTION

Language and its relation to the real-world personalities of someone was explored in various researches, mainly using social media texts with computerized text analysis [1]. Different with previous research [2] [3] [4] that worked on social media data and its metadata, this work used extracted raw instant message from our company-owned customer service contact because the benchmark dataset is not shared anymore. Focusing to use only the text as it is in Bahasa Indonesia (Indonesian Language), we want to add more insight on how to solve APR task in low resource language in addition to the massive user of social media consumer from Indonesia [2].

While evolution and recent work on this topic are focusing on developing a new model, this research experiment on maximizing the utilization of lexical features without any additional data besides the text themselves. Result shows that using spelling correction and fine-tuned pre-trained word embedding on a simple AdaBoost classifier using Gaussian Naïve Bayes could give better result for the classification result on all type of personalities compared to the baseline weighting feature such as Bag-of-Word

II. BACKGROUND

A. Big Five Personality

Similar to other recently published researches, this work presents the individual personalities based on "Big Five" personality traits, also known as OCEAN model: Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. The usage of OCEAN model to the APR task was popular with the *myPersonality* dataset which contain 10,000 *Facebook* status updates from 250 users. This research followed the attempt of solving APR task by making the personalities independent with each other, making that one person could have all of the personality traits or does not have at all.

The first attempt to work on the link of personalities traits and someone's social network activity was using *Facebook* status update and the profile [5]. Throughout the time, there are several approach to solve this task, including having more features from the text, like LIWC (Linguistic Inquiry and Word Count) [1], User's social network metadata like follower / following in twitter [2] [3], using language cues [6] [7], and Ontology modelling [8]

Getting the label for each collected chat were another obstacle in this research. Previously, there are no record of personalities in OCEAN model of our customer. To make this APR task as a supervised classification task, we relied on TIPI (Ten Item Personality Measure) [9] that briefly measure the Big Five dimensions of personalities.

B. Word Normalization

Bahasa Indonesia has rich of morphology with mix culture of traditional native language such as Javanese and the urban style of language that commonly used in metropolitan. There are also some standard use of abbreviation and slang words. However, there are still some nonstandard abbreviation that commonly discovered in the social-media text data.

The idea to put the spelling correction for this task was based from excessive amount of token that occurs only one time in the text. We were aiming to solve the nonstandard abbreviation and reduce the number of unique words occurs in the text. For solving these issues, we combine 2 spelling corrections methodology that complete each other and aware with the contexts on particular phrase window.

- *Dictionary Based*

  The usage of look-up dictionary for solving the abbreviation is relatively one of efficient way to do the word normalization, particularly for the word that has more than three edit distances. This is also more computation friendly to handle the nonstandard and slang words that commonly occurs. We collect top 5323 words for the dictionary with combination from [10] [11] also our attempt to find the domain specific related to our business vocabularies. The samples of the words that are listed in look-up dictionary could be seen in Table 1.

978-1-7281-3333-1/19/$31.00 ©2019 IEEE 19-20 August 2019, Jakarta & Bali, Indonesia
2019 International Conference on Information Management and Technology (ICIMTech)

TABLE I. EXAMPLE OF PRE-DEFINED DICTIONARY CONTENTS

Unformal Word	Formal	Type	English Translation
ngibul	bohong	Slang words	Lying
trmkasih	Terima kasih	Nonstandard abbreviation	Thanks
nangiiis	menangis	Nonstandard typography	crying
skrg	sekarang	Standard abbreviation	Now
Warpin	Warung pintar	Domain specific	Warung Pintar

- *Probability and 2-Edit Distance*

For solving the spelling error that within 2 edit distances, the spelling correction that utilizing probability of occurrence of a word in a corpus [12] is used. It uses Bayes' theorem for finding the word that has most probability for occurs, given a set of word's corrected candidates. This method relies on a pre-defined word count dictionary to find the formal word.

We build the formal word dictionary based on combination of multiple sources, resulting 65114 words and its frequency. In this list, we include the vocabulary that having affixes in order to prevent the spelling error that happened in the affixed word. We realized that Indonesian Language really depends on the affixes in its word because it does not have any tense such as past, present or future tense in English. Another example, Indonesian Language uses affixes like '*di-'or 'me-'* to show the effect of active and passive word for verb.

C. Word Embedding

The implementation of word embedding in personality detection task has been used in previous research, especially with the deep learning approach [13] [7] [14]. In this work, we also use word embedding for getting the context of each word given in the data. We use Gensim [15] for generating the 2 types of word embedding: Word2Vec [16] , and FastText [17]. The difference between both of them is Word2Vec would learn the representation of a word in relation of neighbor words. FastText in addition, relies on Sub word Information, so that in theoretically it robust with spelling error in the text.

III. METHODOLOGY

A. Dataset and Feature Engineering

Dataset that being used for this research are randomly collected from instant messaging WhatsApp from Warung Pintar Customer Service (CS) Team and its customer. There are 22684 lines of raw extracted messages from 133 customers and 3 Customer Services. For this work, we omit the messages from our three CS because we want to focus on finding model for our customer personalities, resulting 11262 lines left as the dataset. Although we omit them, we included our CS messages to the word embedding generation so that the context of each words occurred could be known.

In this work, we proposed 3 different methods and try to look for the best result from it. The previous work for this task is implemented in a social media text in which a serial of text from same subject doesn't have to be connected between each other. Yet, we were using instant messaging text, that it is a continuous stream of text. On one hand, there are several people that the number of texts sent to our Customer service is plentiful, but on the other hand, there are people that sent 1 or 2 sentences only responding what our CS sent to them.

In order to tackle that previously mentioned challenges, these are the 3 feature-engineering for extracting the information from multiple-line raw messages:

1) Word Vector Addition

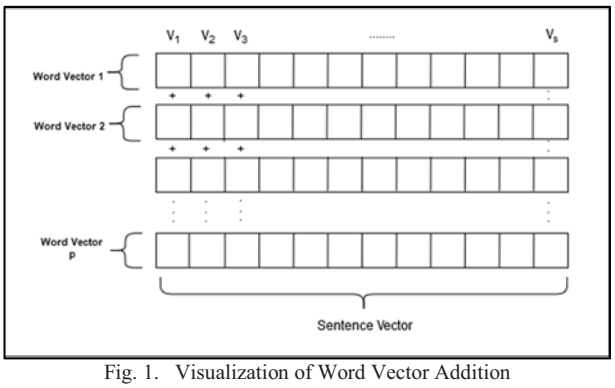

Fig. 1. Visualization of Word Vector Addition

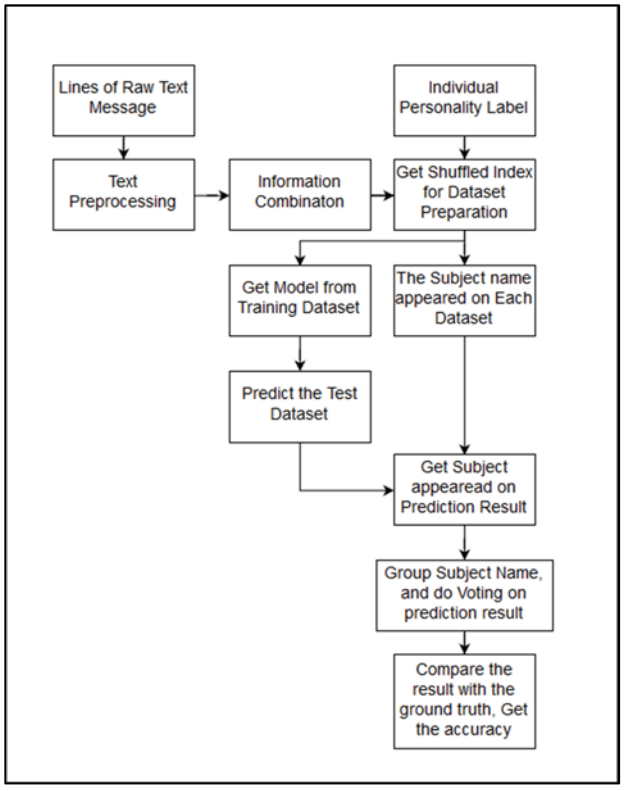

Fig. 2. Flow diagram of first approach

Each word in the data would firstly would be converted into word vector using the word embedding. Then to do information combination, the vector values in position *i* in every word vector are summed into vector values of position *i* in the sentence vector before being divided by the number of words in that text. This approach would generate a sentence vector with length of *n,* following the dimension of the word embedding. The visualization of word vector addition is on figure 1.

We notice that some of the research on previous work used the text dataset and did the personality classification task without considering the similarity of prediction result

of one same person in different text given. We do voting algorithm on prediction result based on the subject that appear on the test dataset. After that, those voting result is compared with the ground truth. The diagram of our workflow in this approach could be seen in figure 2.

2) Sentence Vector Addition

The second approach, we concatenate each text that belongs to same customer. The action squeezes the dataset from 11262 data into only 133 data based on the number of customers. After concatenating, we do the addition of word vector similar in the first approach. In the training phase, we also divided the dataset into 80:20 ratio of train and test data.

Different with the previous attempt, we only do the accuracy checking to the prediction result because it has predicted the personality on full-text of a person's message. Figure 3 illustrates the steps in this approach.

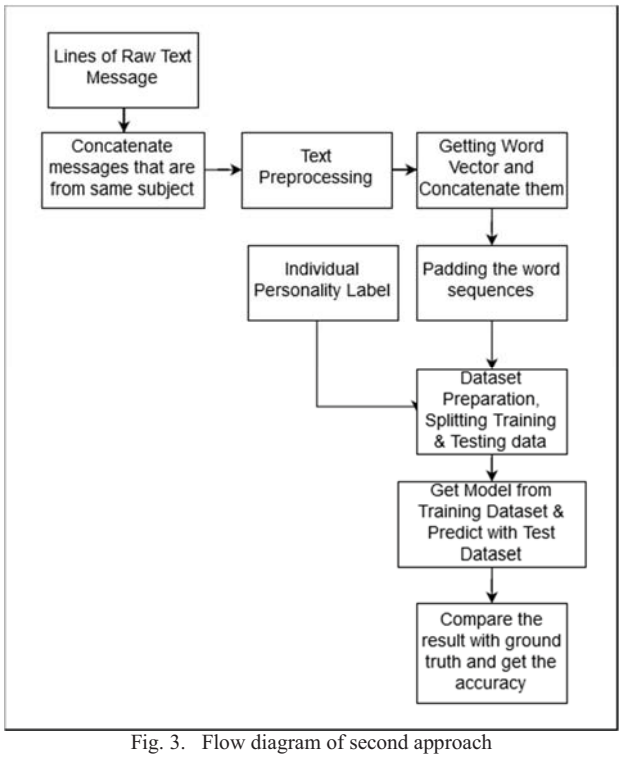

Fig. 3. Flow diagram of second approach

3) Max Pooling from Word Vector Concatenation

The third techniques, we also concatenate each text with the same customer. However, this time we concatenate the word vector of a given word, into a pre-defined length. In this works, we found that a vector of length 100 gives the best result in term of accuracy of the prediction. For customer text that has not enough word to fulfill the length, padding of vector zeros has been put as many as the remainder to the length.

Instead of using 2D array of sentence vector as the representation of the sentence, we pool the maximum value of each word vector so that there is only one representation value from each word vector making the vector back to 1D array of value. The visualization of this techniques could be seen in figure 4 below.

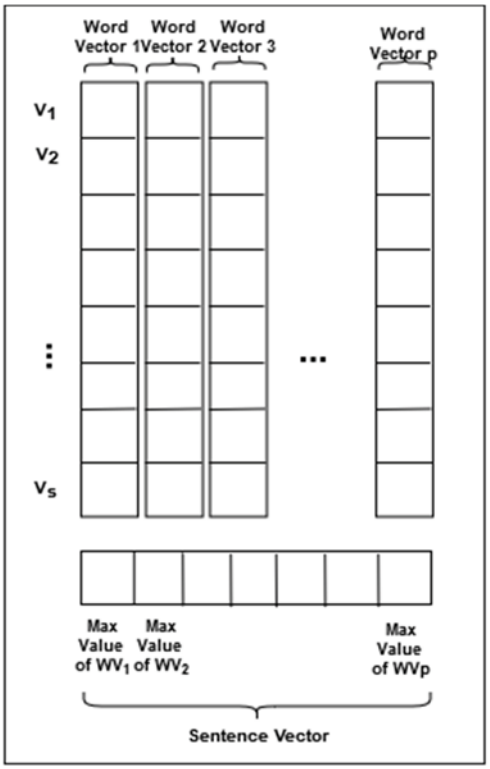

Fig. 4. Pooling the vector value from word vector i

B. Label Dataset

The distribution of label gathered using TIPI Survey, answered by our customer are showed in the Table II below.

TABLE II. TABLE OF LABEL GENERATED FROM TIPI SURVEY

Class	Has Personality	Has Not Personality
Agreeableness	43	90
Neuroticism	72	61
Extroversion	41	92
Consciousness	35	98
Openness	91	42

C. Classification

To see the performance of the feature engineering, we use accuracy as our metrics for the result of classification. It is chosen because we did a binary classification whether a person have or not have a particular personality. The algorithm we choose is AdaBoost Classifier with weak classifier of Gaussian Naïve Bayes using *scikit-learn library* [18]. The formula of AdaBoost is:

$$F(x) = (\sum_{m=1}^{M} \theta_m f_m(x)) \quad (1)$$

Where:
Fm = the mth weak classifier
Θm = The corresponding weight

The baseline feature engineering to be compared with our approach is using *bag-of-word* with *tf-idf* weighting method. However, because AdaBoost classifier does not

accept sparse matrix, we use Gaussian Naïve Bayes for the baseline classification

IV. Result and Discussion

In this section, the performances of our approach for feature engineering the text into various word vector are presented and discussed. The testing was conducted using 5-fold cross validation.

A. Result and Discussion

TABLE III. CLASSIFICATION RESULT USING WORD2VEC WORD EMBEDDING

Personality	Baseline	Word2Vec Embedding		
		Method 1	*Method 2*	*Method 3*
Openness	70	46	52	41
Consciousness	70	44	78	44
Extroversion	52	47	63	70
Agreeableness	66	55	63	52
Neuroticism	52	54	59	41
Average	62	47.2	**63**	49.6

TABLE IV. CLASSIFICATION RESULT USING FASTTEXT WORD EMBEDDING

Personality	Baseline	Fast text Embedding		
		Method 1	*Method 2*	*Method 3*
Openness	70	53	59	63
Consciousness	70	54	74	67
Extroversion	52	47	56	52
Agreeableness	66	47	44	59
Neuroticism	52	52	56	41
Average	62	50.6	57.8	56.4

Table III and Table IV shows the classification result from the test dataset. The highest performance of classification was in the Consciousness personality task, using method 2 and Word2Vec word embedding. It achieved 78% of accuracy. Method 2 was also slightly higher than the baseline model with 1% difference in the average score.

Method 1 and Method 3 gave worse performance than random guessing, the average accuracy for all personalities are below 50%. It could happened because in the method 1, the prediction of each lines of message are having possibility to follow each other. For example, if there are 3 lines that having only one text *"okay"* but having different personalities, the model would be hard to capture the pattern of word vector. Then, this error would be cumulated when the voting happened during the next stage of test.

Although method 3 using FastText embedding could gave result that exceed the random guessing, there is still any personality (Neuroticism) that having accuracy worse than guessing, so we would not recommend to do this approach for doing text classification.

V. Conclusion and Future Work

There are still a lot of rooms for improving and experimenting an Automatic Personality Recognition task. We work on how this classification works in instant message, like chat text in *WhatsApp,* that having no other beneficial features like social media status or metadata did. In the experiment, we have conclusion that concatenating all of the text from same person, then doing information combination by averaging every word vector by its position in the word embedding gave more accuracy then other proposed method.

The future work of this would be feeding more data, in term of more customer collected. Afterwards, given the more than enough amount of data, this task could be represented as a multiclass classification or regression task. The reason behind this is we assume that not everyone does have 100 percent of a personality. They could be based on percentage or scaled from 1-4.

References

[1] Y. R. Tausczik and J. W. Pennebaker, "The Psychological Meaning of Words:LIWC and Computerized Text Analysis Methods," *Journal of Language and Social Psychology,* pp. 24-54, 2010.

[2] G. Yakub, M. H. Tandio, V. Ong and D. Suhartono, "Optimization for Automatic Personality Recognition on Twitter in Bahasa Indonesia," *3rd International Conference on Computer Science and Computational Intelligence,* pp. 473-480, 2018.

[3] V. Ong, A. D. S. Rahmanto, Williem, D. Suhartono, A. E. Nugroho, E. W. Andangsari and M. N. Suprayogi, "Personality Prediction Based on Twitter Information in Bahasa Indonesia," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Prague, Czech Republic, 2017.

[4] M. Hassanein, W. Hussein, S. Rady and T. F. Gharib, "Predicting Personality Traits from Social media using Text Semantics," *13th International Conference on Computer Engineering and Systems (ICCES),* pp. 184-189, 2018.

[5] J. Golbeck, C. Robles and K. Turner, "Predicting Personality with Social Media," in *Proceedings of the International Conference on Human factors in Computing Systems*, Vancouver, BC, Canada, 2011.

[6] F. Mairesse, M. A. Walker, M. R. Mehl and R. K. Moore, "Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text," *Journal of Artificial Intelligence Research,* vol. 30, pp. 457-500, 2007.

[7] N. Majumder, S. Poria, A. Gelbukh and E. Cambria, "Deep Learning-Based Document Modeling for Personality Detection from Text," *IEEE Intelligent Systems,* pp. 74-79, 2017.

[8] A. Alamsyah, M. R. D. Putra, D. D. Fadhilah, F. Nurwianti and E. Ningsih, "Ontology Modelling Approach for Personality Measurement Based on Social Media Activity," in *6th International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2018.

[9] S. D. Gosling, P. J. Rentfrow and W. B. Swann Jr., "A very brief measure of the Big-Five personality domains," *Journal of Research in Personality,* vol. 37, no. 6, pp. 504-528, 2003.

[10] S. Romadhoni, "Leksikon Bahasa Gaul dalam Novel My Stupid Boss Karya Chaos@Work," Yogyakarta, 2012.

[11] A. Purwarianti, A. Andhika, A. F. Wicaksono, I. Afif and F. Ferdian, "InaNLP: Indonesia natural language processing toolkit, case study: Complaint tweet classification," *International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA),* 2016.

[12] P. Norvig, "Norvig.com," February 2016. [Online]. Available: http://norvig.com/spell-correct.html. [Accessed January 2019].

[13] X. Sun, B. Liu, J. Cao, J. Luo and X. Shen, "Who Am I? Personality Detection Based on Deep Learning for Texts," in *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, 2018.

[14] G. An and R. Levitan, "Lexical and Acoustic Deep Learning Model for Personality Recognition," in *Proceeding Interspeech 2018*, Hyderabad, 2018.

[15] R. Rehurek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *The LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010.

[16] T. Mikolov, I. Sutskever, K. Chen, C. Greg and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *International Conference on Neural Information Processing Systems*, Lake Tahoe, Nevada, 2013.

[17] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information," in *arXiv preprint arXiv:1607.04606*, 2016.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duschesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[19] D. Sewwandi, K. Parera, S. Sandaruwan, O. Lakchani, A. Nugaliyadde and S. Thelijjagoda, "Linguistic features based personality recognition using social media data," in *2017 6th National Conference on Technology and Management (NCTM)*, Malabe, Sri Lanka, 2017.

[20] B. Y. Pratama and R. Sarno, "Personality Classification Based on Twitter Text using Naïve Bayes, KNN, and SVM," in *2015 International Conference on Data and Software Engineering (ICoDSE)*, Yogyakarta, Indonesia, 2015.

[21] K. C. Pramodh and Y. Vijayalata, "Automatic personality recognition of authors using big five factor model," in *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, Coimbatore, India, 2016.

[22] D. A. G. Jáuregui, C. Castanier, B. Chang, M. Val, F. Cottin, C. L. Scanff and J. C. Martin, "Toward automatic detection of acute stress: Relevant nonverbal behaviors and impact of personality traits," *Seventh International Conference of Affective Computing and Intelligent Interaction (ACII),* pp. 354-361, 2017.