IT FOR SOCIETY, Vol. 05, No. 01
ISSN 2503-2224

COMPARISON INTENT RECOGNITION ON FOOD DELIVERY SERVICE COMPLAINT IN TWITTER WITH RECURRENT AND CONVOLUTIONAL NEURAL NETWORK

Irfan Nasrullah, S.T.
President University
Bekasi, Indonesia
Irfanasrullah310793@gmail.com

Rila Mandala.
President University
Bekasi, Indonesia
rilamandala@president.ac.id

Abstract--In this research, the case of intent classification for Customer Relation Management (CRM) how to handle complaints as a domain to be followed up, where datasets are extracted from the conversation on Twitter. The research objectives support three key findings to comparing the CNNs and BRNNs model to intent recognition by vectorization text: (1) Which architecture performs better (accuracy) depends on how important it is to semantically understand the whole sequence and (2) Learning rate changes performance relatively smoothly, while the optimal result iterated by change hidden size and batch size result in large fluctuations. (3) Last, how word vectorization is able to define sub-domain of the complaints by word vector classification.

Keywords—complaint, intent classification, CNN, BRNN, fastText

I. INTRODUCTION

Natural language processing (NLP) has benefited greatly from the resurgence of deep neural networks (DNNs), due to their high performance with less need of engineered features. There are two main DNN architectures: convolutional neural network (CNN) (LeCun et al., 1998) and recurrent neural network (RNN) (Elman, 1990). Gating mechanisms have been developed to alleviate some limitations of the basic RNN, resulting in two prevailing RNN types: long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014).

With the explosive growth of information particularly in social media environment, modeling user intent to meet individual user needs is essential. From the user's perspective, understanding user intent could improve the recommendation, personalized search to provide better user experiences. Also from the platform's perspective, a better understanding of user intent could provide accurate products, services to users, so as potentially improve the page view and gross merchandise volume.

Intent classification is an important component of Natural Language Understanding (NLU) systems in any chatbot platform. Intent can help identify actionable information from social media. On this research, intent developed to tracking complaints event in social media. Intent classification (focused on future action) is a form of text classification. Posting short messages (i.e., tweets) through microblogging services (e.g., Twitter) has become an indispensable part of the daily life for many users. Users heavily express their needs and desires on Twitter, and tweets have been considered as an important source for mining user intents) (Zhao et al., 2014).

Two problems arise with the use of internet communication. First, such datasets miss a lot of terms in the vocabulary to use word embeddings efficiently with vectorization Word2Vec from people conversation even Gensim Twitter has more inclusive library to understand Twitter conversation deeply. Second, users frequently make spelling errors.

II. RELATED WORK

Deep learning methods provide excellent performance in various NLP tasks including sentiment classification, sentence and document representation, and machine translation, etc. Recently, Conneau et al. proposed a VDCNN model using the deep convolution network in sentiment classification. The RNN can capture better long-distance dependency between words in sentences compared with CNN.

Since both the CNN and the RNN have their own advantages, several approaches combining both network structures have been applied to NLP tasks. Lai S et al. (2015) constructed a RCNN model that first used Bi-RNN model to obtain the representation of context, and then performed convolution and pooling operations to produce classification results. Zhou C et al. (2015) proposed the C-LSTM model.

The CNN was used to extract the features of texts, and then a LSTM layer and a soft-max layer were used to obtain the classification results. Attention mechanism is able to capture the importance of features about words or sentences, and it is also widely used with CNN or RNN in many NLP or CV tasks.

III. METHODOLOGY

A. Vector Embedding of Words

Representations of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks by grouping similar words. Well-known neural methods such as word2vec and GloVe enable us to obtain distributed vector representations of words, overcoming some of the sparsity issues faced by traditional distributional semantics methods.

In this research, fastText is used as word embedding that over through the limitation of the word2vec on out of vocabulary even though the model result cost more storage. The background of the corpus when training fastText to create model of the word representation based on Wikipedia Indonesia that possibly some term in English will not recognize in word vectorization. Despite, the model of the word embedding still recognize because fastText accommodates that limitation, but the recognize result would be in low accuracy due to the corpus limitation to Wikipedia Indonesia not from another language such as English Wikipedia.

B. Recurrent Neural Network

Recursive neural networks (RNN) are neural networks in which the same set of weights is applied recursively over a graph describing a particular structure. They can be applied on arbitrary structures, by using fixed size representations of variable-length input elements.

Dropout can be applied between layers of the neural network. Dropout is applied by adding new Dropout layers between the Embedding and LSTM layers and the LSTM and Dense output layers in different experiments. In first experiment, the model gets the accuracy of 87.17% that indicate a slightly slower trend in convergence compared to the simple LSTM case.

Figure 1 presents an example of structure (a tree) on which a RNN is applied. We define $[x_1\ x_2]$ the concatenation of vectors x_1 and x_2. x_1, x_2 and x_3 are three input vector $\in R^d$ (where d is the input dimension).

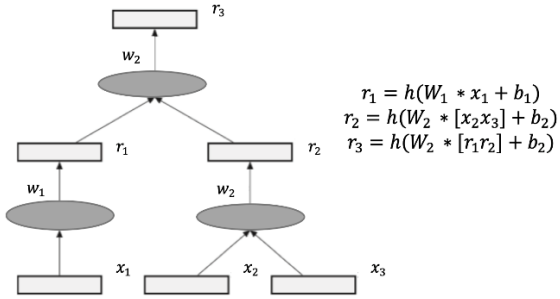

Figure 1 Recursive neural network applied on a tree structure.

The equations r_1, r_2 and r_3 are three vectorial representations of tree nodes. W_1 is a matrix of weight of size $d \times d$. W_2 is a matrix of weight of size $2 \cdot d \times d$. The output of the network is obtained by computing all the node representation recursively, using their child's representations as input. This implies that the node computation order is imposed by the structure of the graph.

C. Intent Classification

Broder first proposed the taxonomy of user's intent as navigational, informational and transactional. Navigational queries intent to reach a particular website. The challenging when facing user intent understanding and modeling:

a. User intent is not always explicitly expressed. For example, on social media, people may discuss various topics such as what they do, how they feel and where they are. And often there're some intents behind these expressions.
b. User intent is something like knowledge graph, or ontology, which is not always available for many domains.
c. User intent is changing. Different people have very disparate interests. And even for one individual, his/her interest is always changing over time.

D. About Data

The data extracted from Twitter API with the key query are the account @Grab_ID and @gojekindonesia then do preprocess by filtering related to "food" services as the domain limitation in this research from all mention and tweets replies. Tweets replies from both accounts (Grab and Gojek) indicate as a response to any complaints related to food services. In other words, anyone that mentions with "food" related to complaints will be recognized as the complaint intention specifically to any sub-domain services problem, for example, food courier, restaurant, amenities, platform, etc. The data attribution that used in this research is the username

and tweet itself as the main data to proceed to intent recognition and classification.

E. Data Preparation

For every dataset--called source data and target data, which source data is a dataset from the user as complaint domain data. Then, target data is output respond from the account usually respond come from customer relation unit the tweet is from the official account in this research both @gojekindonesia and @Grab_ID. And the transfer learning process is as following steps:

a. Filter the data base on keywords "food" as research limitation on food delivery service.
b. Vectorize the dataset on tweet section with word2vec.
c. Build a classifier using the labeled data from the source data.
d. Also, apply this classifier to target data.
e. Perform a feature selection based on particular domain and subdomain.
f. Use the selected feature set label from source data and target data based on domain.
g. Use the two new classifiers together on target data.

F. Bidirectional Recurrent Neural Network

In the bidirectional recursive neural network form, a compositional function (i.e., network) combines constituents in a bottom-up approach to compute the representation of higher-level phrases (see figure above). As a variant of LSTM, words are represented by both a matrix and a vector, meaning that the parameters learned by the network represent the matrices of each constituent (word or phrase).

a. An end-to-end approach to sequence learning that makes min-assumptions on the sequence structure.
b. A layered of Bidirectional Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and then another deep LSTM to decode the target sequence from the vector.
c. The LSTM did not have difficulty on long sentences.
d. The LSTM also learned sensible phrase and sentence representations that are sensitive to word order and are relatively invariant to the active and the passive voice.
e. Reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly, introduced many short-term dependencies between the source and the target sentence.

G. Convolutional Neural Network

Convolutional neural networks initially were used for video recognition tasks, but recent works showed that CNN can also be applied to NLP (natural language processing). Social media and networks become very interesting topic for scientists, since nowadays more and more people are sharing their opinions on different subjects online.

In twitter term, the word representation in the model architecture is a modification of the CNN architecture. Input is a tweet itself, which is representing by a matrix of real numbers, each column is a word of the tweet. The quantity of rows corresponds to a dimensionality of the used word embedding.

CNN has one convolutional layer, one max-pool layer and a full-connected layer with a non-linear function. This network's goal is a binary classification task, predict for a given tweet if it is positive or negative even if CNN used in intent recognition and classification.

In twitter text processing could be transform tweet into vector of words $x = (c_1, c_2, \dots c_n)$, where $c_i \epsilon\, R^l (l = 300)$. $x = (c_1, c_2, \dots c_n)$ For a given x model should give an output 1 or 0. Where 1 is a positive class, 0 is a negative one. The convolutional layer has 3 different filter lengths: 3, 4, 5, there are 100 different filters producing unique feature map by each length. Since there is no padding in convolutional layer, output of each filter has different length, hence one max-pool layer just takes one the most illustrious.

In the present work, simple CNN used with one layer of convolution on top of word vectors obtained from an unsupervised neural language model. These vectors were trained by Mikolov et al. (2013) on 100 billion words of Google News, and are publicly available. The initial assumption keeps the word vectors static and learn only the other parameters of the model.

Tabel 1 Comparison of activation functions with train time and accuracy

Activation function	The train time of one epoch	Accuracy (F-score)
Relu	T≃190 seconds	88.30 %
Sigmoid	T≃130 seconds	89.30 %
Tanh	T≃210 seconds	89.40 %

The vector of word representations is trained on top of pre-trained word vectors which are updated during CNN training. In this research, use the publicly available word2vec to vectorize words. During the training phase, the parameters of the CNN model are learned by passing multiple filters over word vectors and then applying the max-over-

time pooling operation to generate features that are used in a fully connected softmax layer.

a. Every filter performs convolution on the sentence matrix and generates (variable-length) feature maps.
b. Then 1-max pooling is performed over each map, i.e. the largest number from each feature map is recorded.
c. Thus a univariate feature vector is generated from all six maps, and these 6 features are concatenated to form a feature vector for the penultimate layer.
d. The final softmax layer then receives this feature vector as input and uses it to classify the sentence;
e. Assume binary classification and hence depict two possible output states.

IV. RESULT

A. Vector Embedding of Words

In the creation process of the language model of intention with FastText on the first attempt, the corpus that used from Wikipedia due to limitation of the compute resources only 10.000 articles gathered to be trained that contained 8M of words then output 300 dimensions with 63.186 words saved in ".bin" file with size 2,55 GB. Then second attempt, the corpus that used from Wikipedia training with all articles with 370.000 articles, but the vector dimension reduce to 100 dimensions with file size is 1,09 GB due to implication of the computes is the time longer but with lower RAM and storage needed.

Technically, the result of the language model of intent we can describe as nearest neighbours word. Based on the CBOW process of the creation of the fastText model of the word vectorization, the CBOW is learning to predict the word by the context and maximize of the probability of the target word by looking at the context of the sentences.

B. Convolutional Neural Network

The CNN were specifically designed to deal with the variability of two dimensional shapes (x,y). Despite little tuning of hyperparameters, even a simple CNN with one layer of convolution performs remarkably well in working with sequential data. We can apply CNN over words or characters. Our embeddings of dimension d for a word of character length L or a sentence of word length L can be written in the matrix M like:

$$M = [\, x_1\, x_2\, x_3 \ldots . x_i \,]$$

The model is trained by embedding matrix from fastText with the dimention of the vector is 300. In this model, three maxpooling 2D layer is used due to sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.

After designing the model, the summary of the total parameters 35.798.758, trainable parameters is 522.758, and non-trainable parameter is 35.276.000 the overall model display on Figure 4.1. Then, the model is trained with ten epoch and batch size 128. After ten epoch, the result shows that accuracy is 0,875 at epoch eighth even value accuracy is on 0,520 with loss 0,849 and loss value is 1,448 as shown at Figure 2 and Figure 3.

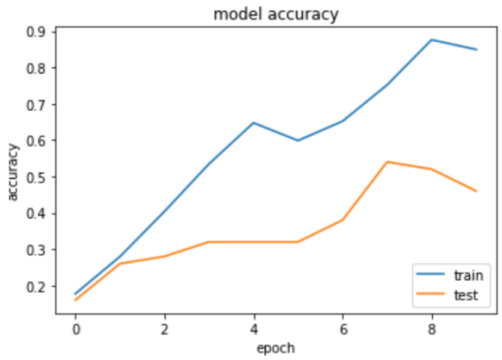

Figure 2 CNN model accuracy

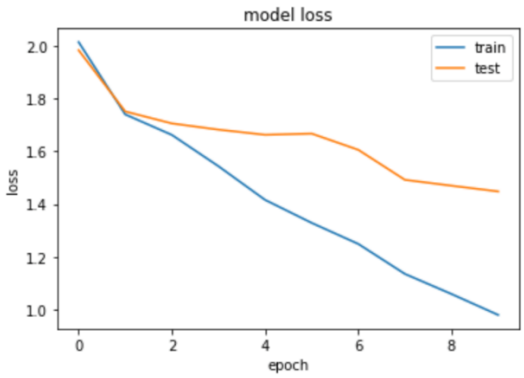

Figure 3 CNN model loss

For optimizing, the researcher plan to tuning hyperparameter and parameter such as num_filter, drop, learning rate (lr), etc. due to limitation of time hyperparameter optimization of the model should develop in future development. Also, limitation of the compute cost the word embedding fastText model should be using larger corpus.

C. Bidirectional Recurrent Neural Network

BRNN is a type of neural network which successfully deals with sequential data. The input is a sequence of (x_1, x_2, ..., x_N) and the output of RNN is a sequence of hidden states (h_1, h_2, ..., h_N). To overcome the limitations of a regular RNN, Schuster (2014) propose a bidirectional recurrent neural network (BRNN) that can be trained using all available input information in the past and future of a specific time frame. BRNN in general, there are two independent LSTM networks in a forward and backward direction that are trained independently (through back-propagation).

Unidirectional LSTM only preserves information of the past because the only inputs it has seen are from the past. The model using bidirectional with parameter value 1024 will run your inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backwards you preserve information from the future and using the two hidden states (h) combined you are able in any point in time to preserve information from both past and future.

After compiling the model, we prepare to train it with an adam optimizer. The training parameter that have set epoch 10 and 512 batch size.

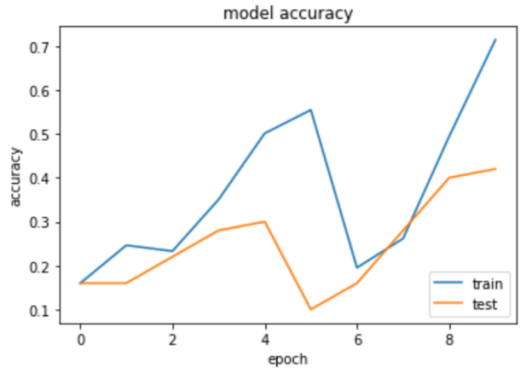

Figure 4 BRNN model accuracy

From the figure 4 shows that model accuracy gained the best model at epoch 10 at 0,714, but the gap from the test data and train data should be notice that the greater number of the gap may caused the model is overfitting. The number of epoch limited to 10 for baseline with the CNN model that success reach the accuracy more than 0,800 in 10 epoch. Despite of the configuration of the training method that only fetch the best epoch, time limitation and compute resources still need to be measured that is the reason to limit compute the RNN only for 10 epochs.

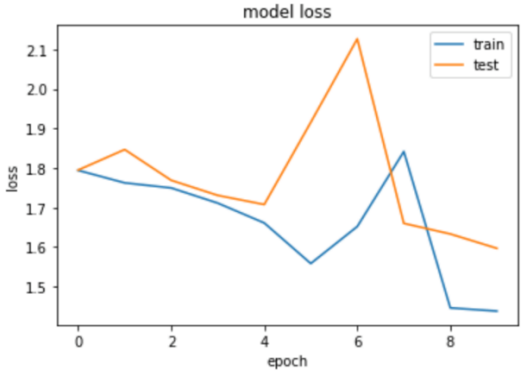

Figure 4.6 *BRNN* model loss

After designing the model, a summary of the parameter such as total parameters 55.530.374, trainable parameters is 55.530.374, and the non-trainable parameter is 0 this is the most optimal model that have created after several trial-error attempt. The hyperparameter tuning objectives is to increase the number of the trainable parameters to avoid the untrainable parameter known as 'dead neural' that may caused bad result. After ten epoch, the result shows that accuracy is 0,714 even value accuracy is at 0,42 with loss 1,438 and loss value is 1,597 overall process by epoch shown in Figure 4 and Figure 5.

V. CONCLUSION

The goal of the research is to develop model of intent based on complaint case that respresent by the keywords "keluhan", "aduan", and "complain" from the Twitter on ride hailing service that have food delivery service. The language model of the intent by those keywords is quite accurate on the second attempt with corpus 370.000 articles of the Wikipedia with 100 dimension vector rather than corpus 10.000 articles of the Wikipedia with 300 dimension. Even though, the greater number of the dimension is better but due to compute and resources limitation better attribute is must be choosen.

The result of the intent of food delivery services complaint by CNN model with precision 0.88, recall 0.87, and f1-score 0.87 with training time for 50 second on 10,762,758 trainable params. The result of the intent of food delivery services complaint by BRNN with precision 0.80, recall 0.81, and f1-score 0.80 with training time for 156 second on 55,530,374 trainable params.

From the comparison of the result above in the context of the intent recognition on food delivery services, the CNN model has better result than

BRNN with number of f1-score CNN is 0.87 and RNN is 0.80 with the training time of the CNN is about 312% faster than BRNN. Despite of the experiment is not focus on training to creates different variables of the hyperparameter, the researcher is keep the CNN and BRNN has similar hyperparameter.

The difference is the CNN has filter size, while RNN not have it because the CNN is a three dimensional training network while BRNN is two dimensional training network. The rest hyperparameter of the CNN and RNN is same such as number filters, drop rate, batch size, and activation on convolutional and bidirectional layer also same is a leaky relu.

VI. RECOMMENDATION

In the future works, before training intent of the words, make sure that word vectorization has the same context with the trained data on deep learning. For example, when training data is from Twitter, make sure we have a corpus of the Twitter, so what taught and predicted has the same word by context. Even though the source of the corpus is limited to Wikipedia, articles from news portal, and Google News that has high degree of formalization, the Twitter could be alternatives for the corpus sources. In Twitter, researchers can scrapping based on keywords with more general in context is essential, since in social media people typing with broad range various of informal text, but if only use Twitter data only is may fail due to Indonesian language is quite similar with Malaysian that implies some of the words has reach context to Malaysian than Indonesian then in vector space of language Malaysian may affect much to Indonesian language.

Since the process of the intent recognition from the deep learning model is only focus on sentence classification, but is still a research challenging to get the best model of the fastText or any other word vectorization. If the best of the model of the RNN or CNN or any other deep learning just firmly discovered then the determination of the word vectorization corpus became the most valuable to any NLU engine for chatbot.

Second, hyperparameter tuning is key to the successful of the implementation of deep learning. So, create different attributes and try different parameters in the layer of deep learning as many as can be explored to get the best model. But, due to time limitation, understanding hyperparameters that affect most to be tuned at least could drive deep learning be better. For future research, there are still any space for improvement such as a sensitivity analysis that creates variables of the hyperparameter training for CNN and RNN since the hyperparameter in both model of deep learning has different implication to accuracy and f1-score, weighting, and with different type of architecture of the deep learning.

Last, there are so many articles that discuss about leaky relu as alternatives from the relu that has alpha attribute as parameter that also can be trained and creates different type of alpha parameters likes hyperparameter tuning. Despite in this research the leaky relu (lrelu) is already used, but in formal discussion such as paper, journal, and indexing as academic articles is still very few and hard to get the accurate articles that discuss only for leaky relu as alternatives to reduce number of the untrainable parameter known as dead neuron in deep learning.

REFERENCES

[1] Cho, K., Gulcehre, C., Chung, J. and Bengio, Y., “Empirical Evaluation of Gated Reccurent Neural Networks on Sequence Modeling”, arXiv preprint arXiv:1412.3555v1, 2014.

[2] Wang, B., K. Liu, and J. Zhao, “Inner attention based recurrent neural networks for answer selection”, In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), 2016, pp. 1288–1297 , Association for Computational Linguistics.

[3] Lai, G. & Chang, Wei-Cheng & Yang, Yiming & Liu, Hanxiao, “Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks”, 2017.

[4] Lai, S., Xu, L., Liu, K., & Zhao, J, “Recurrent convolutional neural networks for text classification”, In:AAAI. Vol. 333, 2015, pp. 2267–2273.

[5] Zhou, C., Sun, C., Liu, Z., & Lau, F., “A C-LSTM neural network for text classification”, 2015, arXiv preprint arXiv:1511.08630.

[6] Mikolov, T., Yih, W., Zweig, G., “Linguistic Regularities in Continuous Space Word Representations”, In Proceedings of NAACL HLT, 2013.