

[CL-Aff Shared Task] Squared English Word: A Method of Generating Glyph to Use Super Characters for Sentiment Analysis

Baohua Sun¹, Lin Yang¹, Catherine Chi¹, Wenhan Zhang¹, and Michael Lin¹

Gyr Falcon Technology Inc., 1900 McCarthy Blvd Suite 208, Milpitas, CA, 95035, US
baohua.sun@gyrfalcontech.com

Abstract. The Super Characters method addresses sentiment analysis problems by first converting the input text into images and then applying 2D-CNN models to classify the sentiment. It achieves state of the art performance on many benchmark datasets. However, it is not as straightforward to apply in Latin languages as in Asian languages. Because the 2D-CNN model is designed to recognize two-dimensional images, it is better if the inputs are in the form of glyphs. In this paper, we propose SEW (Squared English Word) method generating a squared glyph for each English word by drawing Super Characters images of each English word at the alphabet level, combining the squared glyph together into a whole Super Characters image at the sentence level, and then applying the CNN model to classify the sentiment within the sentence. We applied the SEW method to Wikipedia dataset and obtained a 2.1% accuracy gain compared to the original Super Characters method. In this CL-Aff shared task on the HappyDB dataset, we applied Super Characters with SEW method and obtained 86.9% accuracy for agency classification and 85.8% for social accuracy classification on the validation set based on 80%:20% random split on the given labeled dataset.

Keywords: Super Characters · Squared English Word · Text Classification.

1 Introduction

The need to classify sentiment arises in many different problems in customer related marketing fields. Super Characters [8] is a two-step method for sentiment analysis. It first converts text into images; then feeds the images into CNN models to classify the sentiment. Sentiment classification performance on large text contents from customer online comments shows that the Super Character method is superior to other existing methods, including fastText[7], EmbedNet, OnehotNet, and linear models[10]. For example, on the JD_binary dataset[10], which collects Chinese shopping reviews evenly split to positive and negative; the Super Characters method obtained an accuracy of 92.20% while the best existing method obtained one of 91.28%. On another dataset of Rakuten_binary[10],



(a) Raw Super Characters method for English sentence on alphabet level.



(b) Raw Super Characters method for English sentence on alphabet level with changing line to avoid breaking words.



(c) Squared English Word (SEW) method with 6x6 words per image.



(d) Squared English Word method with attention on the first four words.



(e) Squared English Word method using both happy moment text and profiles information, age 36, country India (IND), married (m), male (m).



(f) Squared English Word method using happy moment text and attended profile, age 36, country India (IND), married (m), male (m).

Fig. 1. Demonstrations of raw Super Characters method and Squared English Word method. We use the same example input to illustrate our idea. The raw text input sentence is: “Last month my son got his first trophy in the tennis match and i was very happy and he was very excited to see me his trophy and i took him out for dinner and spend the evening happily with him.”

which collects Japanese shopping reviews evenly split into positive and negative, Super Characters obtained a 94.85% accuracy, compared to the 94.55% accuracy of the best existing method. Yet on another dataset of 11st.Binary[10], which collects Korean shopping reviews evenly split into identifying positive or negative sentiments, the Super Characters method achieved 87.6% compared to best existing method with 86.89% accuracy. The Super Characters method also shows that the pretrained models on a larger dataset help improve accuracy by finetuning the CNN model on a smaller dataset. Compared with from-scratch trained Super Characters model, the finetuned one improves the accuracy from 95.7% to 97.8% on the well-known Chinese dataset of Fudan Corpus.

However, there are a few challenges of using the Super Characters method for Latin language inputs. First, the Super Characters method can be directly applied for Asian languages with glyph characters, such as Chinese, Japanese, and Korean, but not so in such a straightforward fashion to Latin languages such as English. This is because the CNN model connected to the super characters images are designed to recognize two-dimensional images better in the form of a glyph in a square form. Languages like Chinese build their language system upon logograms, which are symbols or characters that serve to represent a phrase or word. If we directly apply Super Characters method to represent sentences in the English language, the Super Characters image is shown as in Figure 1a. Or, as shown in Figure 1b if we try to avoid breaking the words and changing lines because a word is divided between two lines, it will become harder for the CNN model to recognize. In addition, Attention models have succeeded in various fields [2, 9]. How to employ the idea of attention is another challenge for Super Characters, because the second step in Super Characters method is to feed the image of text to a 2-D CNN model for classifying sentiments, but it is difficult to add attention architecture to the CNN model in which Super Characters images are parallel processed.

This paper borrows several ideas from both Super Characters and attention. For the first challenge, we convert each English word to a glyph, such that each word only occupies the pixels within a designated squared area. The resulting algorithm is named Squared English Word (SEW) as shown in Figure 1c. For the second challenge, we add the attention scheme in the first step of Super Characters method, i.e. during the process of Super Characters image generation. In the original Super Characters method, all the text drawn on the image are given the same size, or given the same degree of attention when it is fed into the CNN model connected to it. We add the attention scheme by allocating larger spaces for important words, e.g. those in beginning of each sentence. SEW with attention as shown in Figure 1d. We will describe the details on how to generate these images in the next section.

The CL-AFF Shared Task[6] is part of the Affective Content Analysis workshop at AAAI 2019. It builds upon the HappyDB dataset[1], which contains 10,560 samples of happy moments. Each sample is a text sentence describing the happy moments in English. And each sample has two sets of binary classification

labels, Agency?(Yes|No) and Social?(Yes|No). In this paper, we will apply SEW and SEW with attention on this data set to classify the input texts.

2 Squared English Word method

The original Super Characters method works well if the character in that language is a glyph, and Asian characters in Chinese, Japanese, and Korean are written in a square form. In this work, we extend the original idea of Super Characters [8] by preprocessing each English word into a squared glyph, just like Asian characters. To avoid information loss, the preprocessing should be a one-to-one mapping, i.e. each original English word can be recovered from the converted squared glyph. For text classification task, we propose the SEW method for English sentence input as described in Algorithm 1.

Input: text input: a string of English words
Output: Sentiment Classification Result
Initialization: start a blank image and set the font to draw Super Characters with, set a cut-length of the words, set counter=0, set current_word=the first word in the text input, set the current_word_location for the current_word which is a square area, and get the current_word_area as the area of pixels for current_word_location;
while *not at end of the input text and counter < cut-length* **do**
 get the current_word, set current_alphabet=the first alphabet in the current_word;
 get current_word.length, set location_stepsize=sqrt(alphabet_area) where alphabet_area is current_word_area divided by current_word.length, and set the current_alphabet_location for current_alphabet at the top-left of the squared area of the current_word;
 while *not at end of the current_word* **do**
 draw the current_alphabet at current_alphabet_location;
 move to the next alphabet and update current_alphabet;
 update current_alphabet_location by moving one location_stepsize, or change line if necessary;
 end
 move to the next word;
 counter+=1;
end
Feed into CNN models, such as ResNet-50, and etc.;
return *Sentiment Classification Result*;

Algorithm 1: Super Characters with SEW

The proposed SEW method has shown accuracy improvement on DBpedia dataset provided in [11], as shown in Table 1. DBpedia is a text classification dataset crawled from Wikipedia. It has 14 ontologies, each having 40,000 labeled text in training and 5,000 in testing.

Model	Accuracy
SC[8]	96.2%
SEW (this paper)	98.3%

Table 1. Results of our Squared English Word (SEW) method against original Super Characters (SC) method on DBpedia [11] data set. It shows the effectiveness of the squared method which improves performance by 2.1%. The cut-length is set at $14 \times 14 = 196$ in order for the input with different length to fit. The CNN model used is SE-net-154[5].

Compared to the original Super Characters method, SEW encapsulates one English word per square, rather than one English letter per square. The first word in the sentence goes in the top left square, and the succeeding words follow sequentially from left to right, proceeding onto the next row if necessary. Any remaining space is left empty, as a blank square.

In Figure 1c, the input image consisted of 6×6 squares, and the SEW Super Characters image is generated by only utilizing the happy moment text information. To distinguish from the other approaches below, we call this the SEW-text-only approach.

In Figure 1d, we also introduced an attention-based approach to make our model focus on particular important words or phrases within the input, such as, the first four words of the sentence. By allocating larger sized squares for the Super Characters that would hold certain English words, the convolutional layers within our model naturally dedicate greater emphasis on such words. This is common in the real world when we see signs and emphasized portion is enlarged to take attention as seen in Figure 2. Similarly, people pay more attention to headlines than regular text in newspapers.

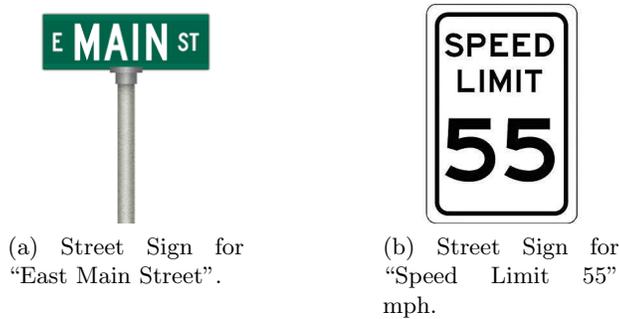


Fig. 2. Street sign example: the enlarged portions of the signs get attention[3, 4].

We call the approach in Figure 1d as SEW-text-only-Attention-Four-words, which applies the attention-based mechanism with an 8×8 input image with

text only information in the happy moment. We chose to teach the network to pay particular attention to the first four words of a sentence, to see if the first four words have a large impact on the overall meaning of the sentence. With this specific implementation of the attention mechanism, we made the first four words two times the size of the rest of the words in the sentence, and positioned it on the center of the image. The regularly sized sentence flows as before, starting at the leftmost square of a row, continuing rightward on all possible places that can contain a squared English word until it hits the rightmost side of the row, then proceeding onto following rows.

In Figure 1e, we also use the profile features and happy moment text together. We set the profile features in Figure 1e as the same size as the happy moment text information. Therefore, the resulting image is a combination of raw text input of happy moment and user-provided profile information. We call this approach as SEW-text-only-and-Profile-Features.

In Figure 1f, similarly, we use both the user profile information and happy moment into the Super Characters image. And we also utilize attention scheme for the user profile information. We call this approach as SEW-text-only-Attention-Profile-Features. By using XGBoost [12] variable importance analysis tool, the parenthood information was determined to be the least important feature in classifying either social or agency when using only the profile information. So we only use four features from profile information, which are age, country, marriage, and gender. For age and country, we use the value as a single word. For marriage, we use initials of category values as the character to draw in the Super Character image, i.e. m (married), d (divorce), s (single), p (separated), w (widow), 0 ("nan"), and leave it empty for empty items. Similar for gender, f (female), m (male), o (other), and N ("nan").

3 Experiments

3.1 CL-AFF Shared Task One

We focused on the above mentioned six approaches as illustrated in Figure 1 for training 2D-CNN models that could discern the agency and social tags of a given happy moment.

For each approach detailed, we trained models by labeling the images with respect to social and agency values. Two separate datasets were created for the training of two different models.

We randomly split the given labeled data into train and test at a ratio of 80%:20%. The histogram of word length distribution is given in Figure 3a for CL-AFF Train dataset, and Figure 3b for CL-AFF Test dataset.

The statistics of training and testing data set are given in Table 2.

Based on the statistics above, we set the cutlength at 36 for SEW-text-only as in Figure 1c. For the 1.39% of the sentences in the labeled data that contained more than 36 words, the 37th word and onwards were not included in the input image. In the shared task 170k test data set, 1.91% were not included.

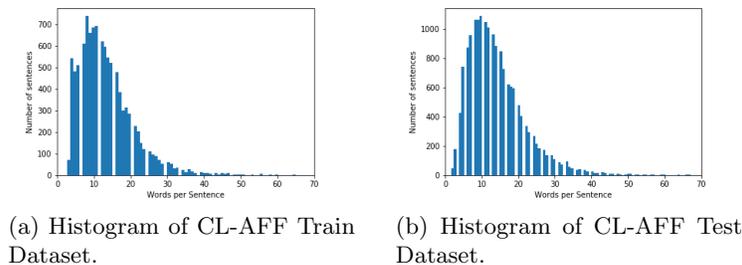


Fig. 3. Histogram of CL-AFF Dataset.

	median	median	max	percentile of 25	percentile of 64
Train	13.44	12	70	92.76%	99.95%
Test	14.28	13	138	90.86%	99.84%

Table 2. The statistics of training and testing data set.

For the attention method, we predefine an 8x8 two-dimensional array to act as the blueprint for the image inputs. There are 0s on all locations that are not designated for the special attended words, to indicate the positions allocated for such words. Of the space reserved for the attended words, all the values are -1 except for the top left box, which is of value 1. Then, we will iterate through every square on the input image. If the corresponding value on the blueprint array based on the given indices a 0, we will draw the next English word in the input sentence using the SEW method. Should the value be 1, we will draw the SEW words in a larger font, and if it is -1, we will skip this iteration of the loop.

Table 3 shows our result based on a split of labeled data into 80%:20% for training and validation. The 2D-CNN model used are all SE-Net-154 [5].

Approaches	Agency Accuracy	Social Accuracy
Raw Super Characters method	85.30%	82.50%
Raw Super Characters method change line	85.70%	83.30%
SEW-text-only	85.90%	83.30%
SEW-text-only-Attention-Four-words	86.00%	83.30%
SEW-text-only-and-Profile-Features	86.30%	85.60%
SEW-text-only-Attention-Profile-Features	86.90%	85.80%

Table 3. Results of raw Super Characters method and Squared English Word (SEW) method on HappyDB data set. For SEW method, attention scheme and profile information are also added. The model used are SE-net-154[5]. Text only means use only happy moment text information. User profile information includes age, country, marriage, and gender.

Comparing SEW-text-only with the raw Super Characters method with line change, we see a little accuracy improvement on agency prediction, from 85.7% to 85.9%. Social accuracy improved from 82.5% to 83.3% compared with raw Super Characters method without line change, although there is no improvement if comparing SEW-text-only to the raw Super Characters with line change. Although we did not see significant accuracy improvement by using SEW method in this data set, it did help improve accuracy by 2.1% for the Wikipedia dataset as shown in Table 1. The main reason for no significant accuracy improvement on this CL-Aff shared task data, is because the data size is not big enough. The CL-Aff data only has a total of 10,560 training samples for different categories, whereas the Wikipedia data set has 560,000 samples for training. Since the generated SEW Super Characters images are fed into CNN models to train, significant accuracy improvement will be observed for large data set because larger data sets help train better CNN models.

For SEW-text-only and SEW-text-only-Attention-Four-words, we see 0.1% accuracy gain on agency label prediction by using attention in this data set, and we see no improvement for social prediction. Using other words to focus instead of using only the first four words, may further improve the accuracy. For example, we can use third party tools to extract keywords related to social or agency, then emphasize these words by enlarging them in the SEW image. Also, for a person’s profile information, like age, country, marriage, and gender, the approach of SEW-text-only-Attention-Profile-Features embed them in the attention area, e.g. put the gender, marriage and etc. information in the attention area.

The significant accuracy improvement for social prediction occurs when we add profile features into the SEW Super Characters image, which jumps from 83.3% to 85.6%. And the agency prediction accuracy also improves from 86.00% to 86.3%. After we further put these profile features into attention, it improves accuracy for both Agency and Social predictions. SEW-text-only-Attention-Profile-Features approach gives the best accuracy of 86.9% for agency prediction, and also the best accuracy of 85.8% for social prediction.

4 Conclusion

This paper borrows several ideas from Super Characters and attention, and we created a squared glyph for each English word. This Squared English Word (SEW) method can be trivially applied to other Latin languages. We apply SEW to this CL-Aff dataset, with user profile information and attention scheme added, we achieved 86.9% accuracy for agency prediction and 85.8% accuracy for social prediction on the given labeled dataset with a split of 80%:20% for training and testing. Pretrained model on large dataset could further improve the accuracy performance by finetuning the CNN models with the relatively small dataset given in this shared task.

References

1. Asai, A.; Evensen, S.; Golshan, B.; Halevy, A.; Li, V.; Lopatenko, A.; Stepanov, D.; Suhara, Y.; Tan, W.-C.; and Xu, Y. 2018. Happydb: A corpus of 100,000 crowd-sourced happy moments. In *Proceedings of LREC 2018*. Miyazaki, Japan: European Language Resources Association (ELRA).
2. Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
3. Digi. 2018. Speed limit 55 mph sign - aluminum sign 8x12. image from web search.
4. HallSigns. 2018. 6 inches extruded aluminum street name sign. image from web search.
5. Hu, J. 2017. Senet-154. github and model download: <https://github.com/hujiefrank/SENet>.
6. Jaidka, K.; Mumick, S.; Chhaya, N.; and Ungar, L. 2019. The CL-Aff Happiness Shared Task: Results and Key Insights. In *Proceedings of the 2nd Workshop on Affective Content Analysis @ AAAI (AffCon2019)*.
7. Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.
8. Sun, B.; Yang, L.; Dong, P.; Zhang, W.; Dong, J.; and Young, C. 2018. Super characters: A conversion from sentiment classification to image classification. *EMNLP2018 workshop WASSA2018*.
9. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 6000–6010.
10. Zhang, X., and LeCun, Y. 2017. Which encoding is the best for text classification in chinese, english, japanese and korean? *arXiv preprint arXiv:1708.02657*.
11. Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 649–657.
12. Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.