Using Skip-Gram Model to Predict from which Show a Given Line is

Dina Chen[1] , **Tom Viering**[1] , **Arman Naseri Jahfari**[1] , **Stavros Makrodimitris**[1]
[1]TU Delft
{

Abstract

Text classification has a wide range of usage such as extracting the sentiment out of a product review, analyzing the topic of a document and spam detection. In this research, the text classification task is to predict from which TV-show a given line is. The skip-gram model, originally used to train the Word2Vec sentence embeddings [Mikolov et al, 2013], is adapted to determine the likelihood of occurrence of a sentence in a TV-show. Based on this feature, a classifier is built to perform the task of this research. The results of the cross-validation show that it reaches an accuracy of 58% when running on the transcript data of 3 shows and 43% on 4 shows, while the accuracies of random guessing are supposed to be 33% and 25%. The difference between the neural networks and the skip-gram model becomes smaller when more shows are added to evaluate the model. Among each 5 fold cross-validation of the two models, the best results appear in the midmost iterations.

1 Introduction

Text classification is a popular area in Natural Language Processing. There are several popular sentence-level classification tasks, such as sentiment analysis for product reviews[Kim, 2014] to examine whether a comment is positive or negative, as well as to examine if a comment is objective or subjective using convolutional neural networks[Wiebe, Riloff, 2005], topic analysis and spam detection. Some studies have been done for text classification on document level, in which the researchers embed the document using the sentence embeddings and perform sentiment estimation and topic classification[Wang, Manning, 2012].

This research examines the performance of the skip-gram model, which is an efficient method to learn vector representations of words[Mikolov et al, 2013] in a text classification task, in which the model predicts from which chosen TV-shows a given sentence is. The 4 shows that are used to evaluate the model are *Friends*, *How I Met Your Mother*, *The Big Bang Theory* and *Modern Family*. The original soft-max function and the objective function of the Word2Vec skip-gram model are adapted to calculate the likelihood of the occurrence of a given sentence in a show. Then the sentence is classified to the show by which it has the highest likelihood. The same task is also performed using the logistic regression neural networks in order to compare with and evaluate the skip-gram model.

Some observations are made during the experiments, such as removing stop-word leads to a better performance and the results of all the cross-validations have a similar shape.

The result of 5 fold cross-validation shows that the skip-gram model and the neural networks have a similar accuracy at their best iteration, but the neural networks perform more stably through all the iterations, while the skip-gram model shows a larger difference between its best and worst results.

In the later sections, the skip-gram model of Word2Vec will be introduced in Section 2 and the adapted version that is implemented in this research will be explained in Section 3. In Section 4, the paper goes through the data pre-processing steps and the experimental procedure. Further, the results of the two methods are presented and discussed in Section 5 and 6. In the end, after Section 7 where the conclusions are made, some approaches that might improve the skip-gram model in text classification tasks are discussed in Section 8.

2 Background of Skip-Gram Model in Word2Vec

The skip-gram model, used to train the vector representations of words, known as Word2Vec, embeds a word by its context, i.e. the words that appear frequently close-by.

The Word2Vec embeddings are trained in the neural networks as follows: given a large corpus of text, each word in the text is represented by a vector, which is initially randomly assigned. The vectors are the parameters to train in the neural networks. In the feed-forward phase, the co-occurence of each word and its surrounding c words are calculated, then during the back-propagation, this probability is maximized.

Given a center word I, the probability of context word O appears close-by is calculated with the soft-max function in [1, eq.1], where the exponential of the dot-product of these two words is taken, then it divides the sum of the exponential of the dot-product of the center word O with all other words in the language.

Knowing how to calculate the co-occurrence of two words, given a center word t and a window size c to indicate the range of the context words, the co-occurrence of each context word and the center word t is calculated and summed up. Taking every word in the language as center word, the same process is done and the average is taken as presented in [1, eq.2], which is the objective function of the neural networks to train the Word2Vec embeddings. This objective is to be maximized during the training process by tuning the vector representation of each word.

$$p(w_O \mid w_1) = \frac{\exp({v'_{wo}}^\top v_{w1})}{\sum_{w=1}^{W} \exp({v'_{wo}}^\top v_{w1})}(1) \tag{1}$$

$$objective = \frac{1}{T}\sum_{t=1}^{T} \sum_{-c \le j \le c, j \neq 0} \log p(w_{t+j} \mid w_t) \tag{2}$$

3 Use Skip-Gram Model to Predict from which Show a Given Line is

In this section, the general idea of how to use skip-gram model to perform text classification tasks is outlined, and the actual adaptations in the functions are shown and motivated.

General Idea of Implementation

The objective function of the Word2Vec embeddings can be easily adapted to predict from which show the given line is, since this function evaluates the co-occurrence of a set of words in a given “language”, in this case, a TV-show. Therefore the main idea of this implementation is to firstly train a Word2Vec embedding model for each show. To classify a sentence, the co-occurrence of the words in that sentence(likelihood of that sentence to appear) is calculated for each show, then the sentence is classified to the show with the maximum likelihood.

Adaptation in the Objective Function

The objective function[1, eq.1] is adapted in order to estimate the likelihood of a sentence appearing in a given show. Instead of taking each word in the whole vocabulary T, we take each word of the to-classify sentence, denoted as S in our likelihood Function (3) as center word. Instead of choosing a window size c to determine the set of context words, we consider every other word in the to-classify sentence (except from the center word) as context word. In this way we calculate the co-occurrence of the words in a given sentence in a given show, later referred as the likelihood of this sentence belonging to the show.

$$likelihood = \frac{1}{S}\sum_{s \in S} \sum_{j \in S, j \neq S} p(w_j \mid w_s) \tag{3}$$

Adaptation in the Soft-max Function

The soft-max function stays the same except from taking the absolute value instead of the exponential of the dot product of two words (see Function 4) to avoid the numerical issues.

$$p(w_O \mid w_1) = \frac{abs({v'_{wo}}^\top v_{w1})}{\sum_{w=1}^{W} abs({v'_{wo}}^\top v_{w1})} \tag{4}$$

The Balancers

Before comparing the likelihoods of the shows and taking the maximum, a step called balancing is implemented, in which the likelihood of each show is multiplied by a factor called the balancer. This is due to the unequal amount of words in each show’s vocabulary. In the denominator of the soft-max function (see Function 4), the dot products of the center word and all the words in the vocabulary are summed up, which means that if a show contains more words, the likelihood calculated by the the soft-max function would be relatively small. Therefore we need to balance the likelihoods before comparing them.

The balancer of a show A is calculated as stated below (Function 5), we calculate the proportion of the amount of word in show A of all shows, take its mean deviation as the percentage by which the likelihood should increase or decrease.

$$balancer_A = 1 + (\frac{\#words_A}{\#words_A llShows} - \frac{1}{\#shows}) \tag{5}$$

4 Experimental Setup

This section introduces the way the data is pre-processed and how the experiments of skip-gram model and logistic regression model are run.

Data Cleaning

The original data was the transcripts of four series: *Friends* , *The Big Bang Theory*, *How I Met Your Mother* and *Modern Family* in hypertext markup language format. To clean the data, the html-tags, location information and other non-speech information are removed to obtain only the dialogues.

Skip-Gram Model to Classify Sentences

In order to perform the skip-gram model, the Word2Vec embeddings are trained for each show using an open-source library Gensim, so that every word has different vector representations in different show.

To classify a sentence, we retrieve the vector representations of all the words in all shows. For each show we calculate the likelihood(Function 3) for the given sentence to appear. The sentence is classified to the show with the maximum likelihood.

For each experiment, 80% of the sentences is used to train the Word2Vec embeddings, the untouched 20% is considered as the test set. The accuracy of the skip-gram model is calculated by the percentage of the corrected classified sentences among all test data. On each data set, 5 fold cross-validation is performed and the confusion matrix is generated.

Logistic Regression Neural Networks to Classify Sentences

In order to evaluate how well the skip-gram model performs by predicting from which show a given line is, the logistic regression neural networks are used to perform the same task.

In order to be fed into the neural networks, the long sentences are cut into short ones to prevent excessive memory usage and the sentences containing less than 3 words are left out. The sentences are transformed into vectors using ELMo (Embeddings from Language Models), a new type of deep contextualized word representation brought out in 2018 by Peters et al[5].

There is a machine learning library – Scikit Learn where we can feed in the 80% of the sentence embeddings of the transcript data to train a classification model, then test on the rest of the data to calculate the accuracy. For this approach we also performed the 5 fold cross-validation

5 Results

This section confirms the effect of the balancers and the stop-word removel in the skip-gram model. Then an observation of the shape of the results is presented. In the last section the general results of the skip-gram model and logistic regression model are presented.

Effect of the Balancers

To confirm the effect of the balancing step, a 5 fold cross-validation on 4 shows is run again without the balancers. The result shows that the accuracy of each iteration drops with 2.43%, 2.80%, 1.58%, 4.46% and 1.48% , and the average accuracy decreases from 42.75% to 40.00% when running without the balancers. This result confirms the effect of the balancing process.

To be more detailed, without the balancing process, the *Friends* class, which contains much less words than the other shows, becomes very dominating, for example: as shown in Table 1, when testing on the 14161 lines from *The Big Bang Theory*, there are 10435 lines that have been classified as *Friends*, the same holds for the test set containing 11665 lines from *How I Met Your Mother*. The reason behind this phenomenon is explained in Section 2, under subsection Balancers and it disappears after introducing the balancing process as shown in Table 2.

Without Balancers

		Classifed As		
Lines From	$\#lines$	F	T	H
From F	13059	**11458**	431	1170
From T	14161	10435	**1988**	1738
From H	11665	7553	704	**3408**

Table 1: Confusion matrix of skip-gram model classification without balancing process, trained on the first 80% of the data, tested on the rest. F as *Friends*, T as *The Big Bang Theory* and H as *How I Met Your Mother*

With Balancers

		Classifed As		
Lines From	$\#lines$	F	T	H
From F	13059	**6311**	4283	2465
From T	14161	3044	**8217**	2900
From H	11665	2681	3971	**5013**

Table 2: Confusion matrix of skip-gram model classification with balancing process, trained on the first 80% of the data, tested on the rest. F as *Friends*, T as *The Big Bang Theory* and H as *How I Met Your Mother*

Effect of the Stop-Word Removal

The research of Toman et al [6] suggests that removing stop words but omitting word normalization (stemming and lemmatization) can be the best way to pre-process the data for text classification tasks. In this research, only the stop-word removal is performed.

To examine the effect of the stop-word removal in this research's case, the cross-validation is performed again on the data of 3 shows, without the stop-word removal. The results shows that the accuracy of each iteration decreases with 3.82%, 1.61%, 4.80%, 6.66% and 0.21%. The average accuracy decreases from 57.75% to 54.33%. This result confirms the positive effect of the stop-word removal on the performance in the skip-gram model.

Shape of the Results

Observing each iteration of all the cross-validations that have been performed, it is noticeable that the best result almost always appears at the third or the fourth fold, and the worst result appears at the first or the last fold (Figure 1). The trend presents a shape of a parabola concaving upwards. This holds for the skip-gram model as well as the logistic regression neural networks. The reason for this phenomenon is unknown.

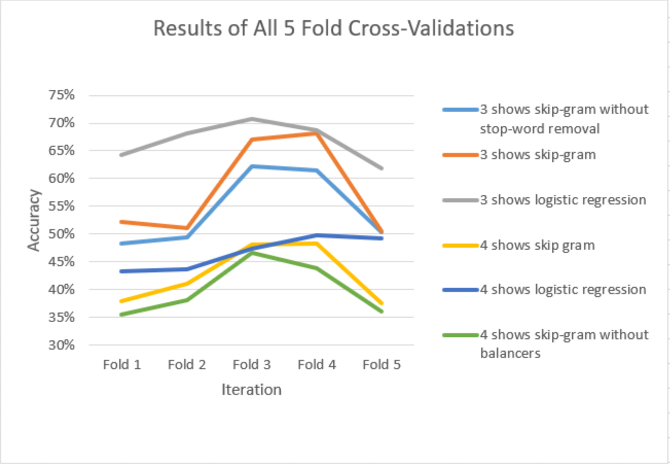
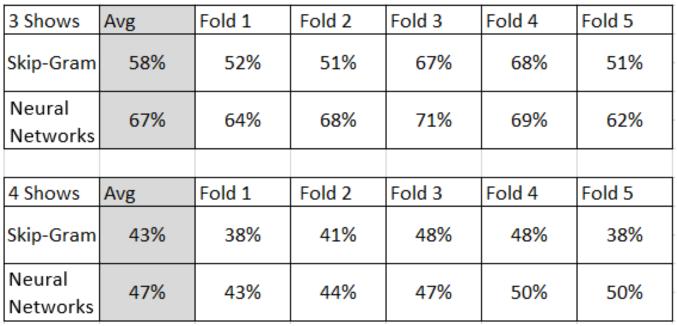


Figure 1: Results of all cross-validations.



3 Shows	Avg	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Skip-Gram	58%	52%	51%	67%	68%	51%
Neural Networks	67%	64%	68%	71%	69%	62%

4 Shows	Avg	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Skip-Gram	43%	38%	41%	48%	48%	38%
Neural Networks	47%	43%	44%	47%	50%	50%

Figure 2: Results of 5 fold cross-validation of both skip-gram model and logistic regression neural networks, run on 3 shows: *Friends* , *The Big Bang Theory* and *How I Met Your Mother*, and 4 shows (with *Modern Family* added).

General Results of Skip-Gram Model and Logistic Regression Model

Both methods are run on 3 shows and 4 shows. When running on the data of 3 shows, we expected an accuracy of 33% by random guessing. The skip-gram model reaches 58% and the logistic regression neural networks achieves 67%. If we perform random guessing on 4 shows, the accuracy would be 25%. The skip-gram model has an accuracy of 43% and the logistic regression neural networks 47%. The statistics about the amount of sentences in each test set and the confusion matrix are shown in Figure 2.

6 Discussion

This section discusses the performances of the 2 methods and describes a remaining problem in the skip-gram model – the dominance of a class.

Comparison of the 2 Methods

The logistic regression neural networks have always been expected to have a better performance in this research than the skip-gram model since it operates on the advanced techniques of artificial intelligence. From Figure 2 we can see that it has a higher accuracy than skip-gram model in each iteration, and its average, when running on 3 shows, reaches an accuracy of 67% while the skip-gram model reaches 58%. The gap becomes smaller when running on the data of 4 shows, with 47% for logistic regression model and 43% for skip-gram model.

We can observe that the highest accuracy that these two models obtain on a single fold doesn't differ a lot (68% and 71%, 48% and 50% in Figure 2), this shows the potential of the skip-gram model to compete with the neural networks. But it is worth noticing that the skip-gram model performs less stably than the logistic regression model through the whole cross-validation, which means that there is a large interval between the best and the worst performance – 17% when running on 3 shows, and 10% when running on 4 shows (Figure 2). This leads to a lower average accuracy for the skip-gram model than the logistic regression model, which doesn't show such large difference between the best and the worst results.

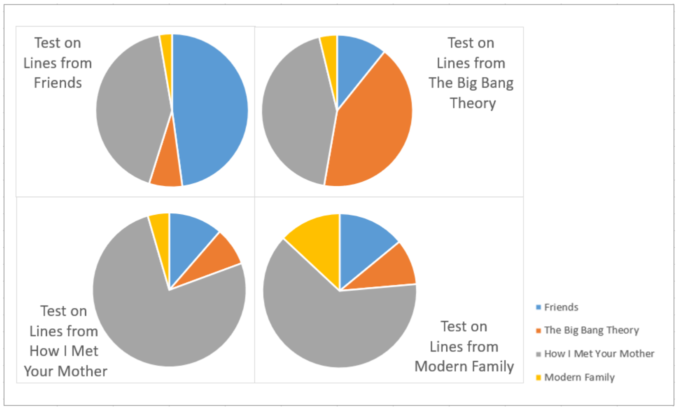


Figure 3: Result of the second fold of the cross-validation. The charts present to which show the lines are classified to.

Dominance of a Class

The problem with *Friends* dominating the results of the skip-gram model was solved by introducing the balancers, but this phenomenon appears again in the cross-validation on 4 shows with *How I Met Your Mother* dominating the results and *Modern Family* being dominated.

Figure 3 presents the result of the second fold of the cross-validation, We can see the grey part, indicating *How I Met Your Mother* takes up a large proportion in all four test cases. Among the test case containing only lines from *Modern Family*, there are much more lines being classified to *How I Met Your Mother* than *Modern Family* itself.

In general, the skip-gram model is least likely to classified a line as *Modern Family*. In Figure 3, yellow part which indicates *Modern Family* takes up a small proportion. The cause of this phenomenon is unknown.

7 Conclusions

In this research we examine how the skip-gram model performs in text classification tasks – given a set of similar TV-shows, predict from which one a given sentence is from.

There are some observations made on the results of the experiments: 1, the stop-word removal leads to better performance in this research. 2, the best result in a cross-validation appears in the third or the fourth fold, and the worst appears in the first and the last fold. This shape of result is consistent in this research.

As for the performance of the skip-gram model: the accuracy of the skip-gram model of it's best iteration in the cross-validation can be near the best result of the logistic regression neural networks. However, the performance varies with a larger margin than the logistic regression model, which leads to a definite lower average accuracy. When running on the transcript data of 4 shows, the overall accuracy of the skip-gram model comes closer to the logistic regression model (43% versus 47%) than when running on 3 shows (58% versus 67%).

8 Future Works

Stemming and Lemmatization

Stemming and lemmatization have been left out in this research because it has been motivated that they may not be a good idea for the text classification tasks in the study of Toman et al [6]. However, there is no unique combination of data pre-processing that guarantees the best performance for every domain and purpose in the text classification problems[UysalGunal, 2014]. As it has been proven that the stop words removal leads to higher accuracy, the stemming and lemmatization can also be introduced in the data-cleaning phase in the future, since they both aim at reducing the noises in the data to train, in order to focus on the actual meaning of the words and not get distracted by the grammatical conjugations.

Weighted Likelihood

Another step that might improve the accuracy of the skip-gram model is taking the word frequency into account when performing the likelihood calculation for each center word. This modification is motivated by the observation that the Word2Vec tends to give large vectors to frequent words, because the co-occurrence probabilities of words are calculated using their dot products[Arora et al, 2016].

Knowing the frequency p(w) of a word w in the vocabulary, the weighted factor, extracted from Algorithm 1 from *A Simple But Tough-to-Beat Baseline for Sentence Embeddings* [Arora et al, 2016], is calculated as follow:

$$\frac{\alpha}{\alpha + p(w)} \tag{6}$$

with the α being a scalar hyper parameter which is fixed to 10^{-3}. Therefore, the objective function will be adapted as follow:

$$\frac{1}{S}\frac{\alpha}{\alpha + p(s)}\sum_{s\in S}\sum_{j\in S, j\neq S} p(w_j \mid w_s) \tag{7}$$

Scaling

We observe that when more TV-shows are used to evaluate the models, the difference between the neural networks and the skip-gram model become smaller. The next step can be adding more shows and see if the performance of the skip-gram model gets further closer to the neural networks and if at some point they will become evenly precise in prediction.

References

[1] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[2] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[3] Wiebe, J., Riloff, E. (2005, February). Creating subjective and objective sentence classifiers from unannotated texts. In International conference on intelligent text processing and computational linguistics (pp. 486-497). Springer, Berlin, Heidelberg.

[4] Wang, S., Manning, C. D. (2012, July). Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2 (pp. 90-94). Association for Computational Linguistics.

[5] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[6] Toman, M., Tesar, R., Jezek, K. (2006). Influence of word normalization on text classification. In Proceedings of the 1st international conference on multidisciplinary information sciences technologies (Vol. 2, pp. 354–358). Merida, Spain.

[7] Uysal, A. K., Gunal, S. (2014). The impact of preprocessing on text classification. Information Processing Management, 50(1), 104-112.

[8] Arora, S., Liang, Y., Ma, T. (2016). A simple but tough-to-beat baseline for sentence embeddings.