# Numeric-attribute-powered Sentence Embedding

Sicong Kuang Brian D. Davison
Lehigh University, Bethlehem PA, USA
{sik211, davison} @cse.lehigh.edu

***Abstract*—Modern embedding methods focus only on the words in the text. The word or sentence embeddings are trained to represent the semantic meaning of the raw texts. However, many quantified attributes associated with the text, such as numeric attributes associated with Yelp review text, are ignored in the vector representation learning process. Those quantified numeric attributes can provide important information to complement the text. For example, review stars, business stars and number of likes, etc., have great influence on interpreting the semantic meaning of text. Numeric attributes associated with the text often reveal the quantity or the significance of the object that the number is modifying. We propose an algorithm using vector projection to generate numeric-attribute-powered sentence embeddings for multi-label text classification. We evaluate our algorithm on a public Yelp dataset, showing that classification performance improves significantly when numeric attributes are incorporated well.**

***Index Terms*—numeric attribute, sentence embedding, classification task**

## I. Introduction

Recent years have seen great success of word embedding or sentence embedding both as features or inputs to sentence-level classification [1], [2]. Researchers usually learn embeddings from the raw text [3], [4]. Many numbers associated with the text are ignored in the learning process. However, those numbers can provide important global information for the interpretation of the text. For example, in a review dataset there are many numeric attributes associated with text. Take the numeric attribute review_star for example. It is usually designed as a 1-5 rating-scale attribute (as in Amazon, Yelp and TripAdvisor). An easygoing user writes a review “it is good” and marks 5 in the review_star attribute while a stern user might mark 3 in the review_star attribute even if he or she uses similar words in the review text such as “not bad experience”. Combining the review text with the value in the attribute review_star, we have a deeper understanding of the significance level of the text (how “good” as in the example) Thus raw text alone can only provide limited information regarding semantics of the sentence.

In this paper we focus on formulating the vector representation of numeric attributes and combine such attribute information into a sentence vector representation for a multi-label classification task. We propose to use vector projection to formulate the vector representation of the numeric attribute. We define the name of the numeric attribute as subject. Instead of ignoring the numbers associated with the text, we incorporate the number by regarding the (number, subject) pair as a whole and treat the number as the scalar projection of the subject. In summary, the main contributions of this paper are three-fold: (1) we are the first to propose the vector projection approach to incorporate numbers associated with the text to formulate vector representation; (2) we propose an algorithm to generate numeric-attribute-powered sentence embedding; and, (3) we evaluate the proposed approach and the algorithm on a multi-label classification task using public data. Our experimental results demonstrate the effectiveness of the proposed approach.

## II. Related work

Sentence embeddings have been built via various methods [5], [6], [7]. Wieting et al. generate sentence embedding based on the supervision from the Paraphrase Database (PPDB) [8]. The authors utilized six sentence embedding models such as recurrent neural network (RNN), deep averaging network (DAN) [9] and LSTM [10]. They also choose a simple model, averaging all word embeddings in the sentence. They embedded the developed model into an objective function. This objective function is a margin loss function based on PPDB. With the additional semantic supervision of the PPDB, the authors expected similar sentences’ trained embeddings should be high in cosine similarity. Their results showed that the most simple model, averaging all word embeddings in the paraphrase had a better result than those complicated models such as RNN and LSTM. Socher et al. build a recursive neural tensor network to combine components in the sentence for sentiment prediction [11]. Researchers have made progress to take the advantage of both categorical features and numerical features for classification task. Zhao et al. found that a decision tree model is good at handling numeric features while a factorization machine is good at handling categorical features, and so proposed a combined model [12].

Researchers have made great efforts to take advantage of numbers in the text in text classification tasks. Macskassy et al. proposed a way to incorporate numbers in the text [13] by converting numbers to bag-of-tokens and incorporating those tokens into text that was represented as a bag of words. Their main contribution lies in the proposed algorithm to optimally split the number tokens such that if two numbers are close, these sets will be similar, and if they are further apart the sets will be less similar. For example the number 1800 could be represented as [“$lengthunder500''$, “$lengthover500''$, “$lengthunder1500''$, “$lengthover1500''$]. However, treating numbers as tokens results in losing their original quantity and value. Using a pre-defined discretization set, numbers such as 499 and 501 are less similar than 499 and 1 in the example. Besides it is not scalable in the sense

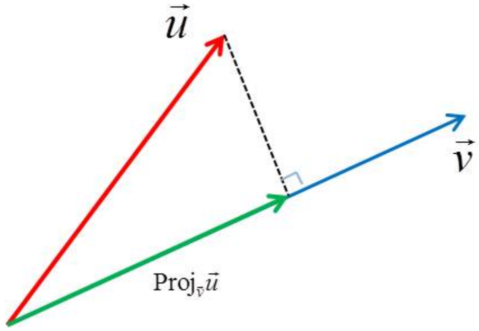

Fig. 1. Projection of $\vec{u}$ on $\vec{v}$, $\texttt{Proj}_{\vec{v}}\vec{u}$ is the projection of $\vec{u}$ onto $\vec{v}$.

that when new training data is pulled in, new split points need to be generated.

Aman et al. extracted number relations in text [14]. Their goal is to extract information from a sentence in the form of a tuple with quantity as the second entity, subject as the first entity and a relationship phrase to describe the relation between the quantity and the subject. They devise two algorithms to extract numerical relations. One is rule-based and the other is a learning system using a graphical model.

Chaganty and Liang [15] tackle a problem of how to automatically generate short descriptions of phrases containing numbers using units or concepts that are easier or familiar to illustrate. Their first step is to manually collect a knowledge base consisting of 9 fundamental units. Second step is to use regular expression patterns to collect phrases containing numbers to form a dataset. Then they use a graph to represent its unit and all the units mentioned in the knowledge, so that when future phrases come they could find all the units close to it for formula representation. After all the formulas are generated based on all the units in the knowledge base, they use crowdsourcing to choose the most appropriate formula through rating (in this way they have labels for formulas in the training set). Their final step is to generate a brief description of the phrase containing a number using a sequence-to-sequence RNN.

However, none of the existing research combines the quantitative nature of the numbers and neural language model embeddings to form sentence embeddings from textual data with numeric attributes.

Multi-label problem has been throughly studied. Gong et al proposed a novel framework for multi-label propagation [16]. It assigned a teacher for each label. The framework utilized the teacher-learner strategy. The teacher assigned simplest labeled samples for the model to learn at the very beginning. According to the metrics' feedback, the teacher gradually raised the bar and gave more complex samples to the model. We use multi-label classification to evaluate our work.

## III. Background of vector projection

We present guidelines when designing an algorithm to generate a numeric-attribute-powered sentence embedding:

1) Numbers alone do not carry semantic meaning. A proposed approach should combine numbers and the subjects they are modifying.
2) Since numeric attributes usually share the same context with the text, numbers and words in the sentence should come from the same vector space to enable the combination of the (number, subject) pair.
3) Since our goal is to use the generated sentence embedding as features for a classification task, the formulated embedding of the (number, subject) pair should account for at least the semantic and the syntactic regularities of the subject alone.

Based on the desiderata above, the vector projection approach is ideal. We use this approach to combine the (number, subject) pair.

Figure 1 illustrates the vector projection approach. We use this approach to combine the (number, subject) pair. It shows the projection of vector $\vec{u}$ on $\vec{v}$ and $\texttt{SProj}_{\vec{v}}\vec{u}$ is the scalar projection of $\vec{u}$ onto $\vec{v}$. In our case, the scalar projection $\texttt{SProj}_{\vec{v}}\vec{u}$ is the number in the sentence; $\hat{v}$ is the unit vector in the direction of $\vec{v}$, the subject that the number is modifying. Our goal is to find the vector representation of the (number, subject) pair, $\texttt{Proj}_{\vec{v}}\vec{u}$. Based on the vector projection approach, $\texttt{Proj}_{\vec{v}}\vec{u}$ is obtained by:

$$\texttt{Proj}_{\vec{v}}\vec{u} = \texttt{SProj}_{\vec{v}}\vec{u} \times \hat{v} \tag{1}$$

## IV. Numeric-attribute-powered sentence embedding

This section introduces in detail our proposed algorithm to incorporate numbers into sentence embeddings by utilizing the simple idea of vector projection. The numeric-attribute-powered sentence embedding consists of two parts: the text vector representation $t_{emb}$ and the numeric attribute vector representation $n_{emb}$. $t_{emb}$ corresponds to pure text vector representation leaving out the numeric attributes associated with the text. $n_{emb}$ corresponds to the vector representation of the numeric attributes. The algorithm 1 illustrates the proposed algorithm to generate numeric-attribute-powered sentence embeddings. Generally we use the vector projection approach described in section III to represent the numeric attributes in vector space. We use the $\chi^2$ test to rank all the numeric attributes, then a holdout set to select the optimum set of numeric attributes to include. The text vector representation $t_{emb}$ is represented as the average of the word representations in the sentence. In general, we calculated $t_{emb}$ and use vector projection approach to represent $n_{emb}$; We concatenate $t_{emb}$ and $n_{emb}$ to form the numeric-attribute-powered embedding.

Recall that our goal is to verify the effectiveness of the vector projection approach to incorporate numeric attributes into vector representation. Thus we do not focus on generating the vector representation of text leaving out the numeric attributes. According to Wieting et al. and Kenter et al., the simplest averaging model is competitive with systems tuned for the particular tasks while extremely efficient and easy to use [8] and it has proven to be a strong baseline or feature across a multitude of tasks [17]. We adopt the averaging model

**Algorithm 1** Numeric-attribute-powered sentence embedding

```
Data: numeric-attribute intense text dataset of size $m$: $D(s_1, s_2, \dots, s_m)$. We divide the dataset into $D_{train}$, $D_{holdout}$ and $D_{test}$, where $s_1(t_1^1, t_2^1, \dots, t_{p^1}^1, n_1^1, n_2^1, \dots, n_q^1)$, $s_2(t_1^2, t_2^2, \dots, t_{p^2}^2, n_1^2, n_2^2, \dots, n_q^2)$, ..., $s_m(t_1^m, t_2^m, \dots, t_{p^2}^m, n_1^m, n_2^m, \dots, n_q^m)$; $t$ is text feature of size $p$; $n$ is numeric attributes of size $q$; each $t$ represents a word in the text; each $n$ represents a numeric attribute consist of $n_{val}$, the number and $n_1$, the numeric attribute name/type; both of them could be multi-dimensional. Result: the selected numeric attributes to incorporate; a matrix of size $m \times (|t_{emb}| + |n_{emb}|)$;
$t_{emb} \leftarrow$ generating_text_embedding($D$)
$\overrightarrow{n_1}, \overrightarrow{n_2}, \dots, \overrightarrow{n_q} \leftarrow$ generate_vector_representation_number_attribute($n_1, n_2, \dots, n_q$)
$\hat{n}_1, \hat{n}_2, \dots, \hat{n}_q \leftarrow$ compute the unit vector in the direction of $\overrightarrow{n}$ by dividing the norm on each element of $\overrightarrow{n}$
$\hat{v}_{n1}, \hat{v}_{n2}, \dots, \hat{v}_{nq} \leftarrow$ vector_projection($n_{val}$, ($\hat{v}_{n1}, \hat{v}_{n2}, \dots, \hat{v}_{nq}$))
$Q \leftarrow$ Chi2($\hat{v}_{n1}, \hat{v}_{n2}, \dots, \hat{v}_{nq}$))    ▷ Q is a list of the sorted numeric attribute using feature selection approach Chi2
curr_index $\leftarrow 1$    ▷ curr_index records the index of the current evaluating size of the ranked numeric attributes
currMetric $\leftarrow 0$    ▷ currMetric records the current metric value in the $D_{holdout}$
$maxMetric \leftarrow 0, numIndex \leftarrow 0$    ▷ maxMetric records the highest metric value in the $D_{holdout}$ so far
    ▷ numIndex records the index of the numeric attributes when maxMetric is achieved
while $C \leftarrow$ top_next_element(Q,curr_index) do
    $X \leftarrow$ concatenate($t_{emb}$,$Q(1:C)$)    ▷ $X$ is the feature set which combines text vector representation
    currMetric $\leftarrow$ evaluation($D_{holdout}$, $X$)
    if currMetric $>$ maxMetric then
        $numIndex \leftarrow curr\_index$
        $maxMetric \leftarrow currMetric$
    else
        $currMetric \leftarrow 0$
$X \leftarrow$ concatenate($t_{emb}$,$Q(1:numIndex)$)  ▷ use numeric attribute features $Q(1:numIndex)$ selected from $D_{holdout}$ to formulate numeric-attribute-powered sentence embedding to generate the test feature set
Return $X$
```

to generate text vector representation $t_{emb}$ before we add numeric attributes to the embedding.

$$t_{emb} = \frac{1}{p} \sum_{i=1}^{p} W_{\omega}^{t_i} \tag{2}$$

Equation 2 shows the averaging model. Considering in a sentence $s$ we have a word sequence $s = \langle t_1, t_2, \dots, t_p \rangle$, $W_{\omega}^{t_i}$ is the word embedding for word $t_i$.

To generate the numeric attribute vector representation, we first find the vector in the direction of the subject that the number is modifying by the numeric attribute name. We use the same averaging model in Equation 2 to obtain the vector representation of the subject $\vec{v\prime}$. Then we obtain $\vec{v}$, the unit vector in the direction of the subject that the number is modifying, by dividing the norm on each element of $\vec{v\prime}$. Then we use Equation 1 to obtain $\vec{u}$, the vector representation for the (number, subject) pair. Since there could be multiple numeric attributes associated with the text, we use feature selection algorithm Chi2 [18] to rank the numeric attributes and use a hold-out dataset to select the optimum set of numeric attributes that should pair with text vector representation. We conduct a snowball approach to select the optimum set of numeric attributes. Namely we first incorporate the numeric attribute that ranks first, then we incorporate the top 2 numeric attributes, then top 3, etc. We use the same averaging model in Equation 2 to generate the numeric attributes vector representation.

## V. EVALUATION

We verify our proposed algorithm on a multi-label classification task. We first describe the dataset. Then we elaborate on the metrics and the experimental result.

### A. Dataset

We evaluate the proposed algorithm on the publicly available Yelp dataset offered in 2017 as part of round 9 of the Yelp Dataset Challenge[1]. Since our proposed method does not involving training embeddings from scratch, we do not need a large corpus. In the experiment we use the pre-trained Word2Vec embedding trained on Google News[2]. The statistics in the experiment are shown in Table I. We decide to evaluate the proposed algorithm on Yelp dataset for several reasons: first, it is a numeric attribute intensive dataset; second, our goal is to predict the business type for each review. Thus the numeric attributes associated with review text such as “review counts” and “review votes useful”, etc., are related to the task and this is also proved in the feature selection chi2 step. Third, this dataset is made for competition and data structure is clear and well-stored. Thus it is easily accessible to extract numeric attributes associated with the review text. The number and the subject pair is clearly stated in the dataset. An example of the numeric attributes associated with text is shown in Table II. There are 7 main business types in the dataset, such as

[1] https://www.yelp.com/dataset_challenge

[2] https://code.google.com/archive/p/word2vec/

| Dataset | Yelp |
|---|---|
| review size | 20000 |
| numeric attributes | 174 |
| labels | 7 |

TABLE I
THE STATISTICS OF THE YELP DATASET

|  | numeric attributes |
|---|---|
| 1 | business review count |
| 2 | user votes useful |
| 3 | user votes cool |
| 4 | user votes funny |
| 5 | review count |
| 6 | friends |
| 7 | review stars |
| 8 | business stars |

TABLE II
EXAMPLES OF THE NUMERIC ATTRIBUTES ASSOCIATED WITH REVIEW TEXT

| F-score | Stratified Classifier | Review Text Only | Numeric-attribute-powered Sentence Embedding |
|---|---|---|---|
| Macro-average | 18.60% | 47.30% | 54.00% |
| Micro-average | 36.90% | 76.90% | 79.20% |

TABLE III
PERFORMANCE OF THE TWO METRICS WITH PRE-TRAINED WORD EMBEDDING DIMENSION 300

Active Life, Food and Restaurants and Services. A review may belong to one or several business types. Thus it is a multi-label classification problem.

### B. Baselines

To evaluate the effectiveness of the proposed algorithm, we compare it with two baselines.

1) stratified classifier: generates predictions by respecting the training set's class distribution.
2) review text as features: in this baseline we only consider $t_{emb}$ as features for the multi-label classification task. We use the averaging model to generate $t_{emb}$. We use the same linear SVM on this baseline and the proposed algorithm.

### C. Experimental Setup and Result

We employ 5-fold cross-validation and divide the dataset into 3 folds for $D_{train}$, 1 fold for $D_{holdout}$ and 1 fold for $D_{test}$. We build a binary linear SVM classifier for each category. The classification performance is measured via two commonly used evaluation criteria, macro-average and micro-average [19]. F1 measure is commonly used for binary classification. Both macro-average F score and micro-average F score are based on F1 measure. Macro-average F score is the arithmetic mean of F1 measure across all categories; thus treating all category equally; micro-average F score is the harmonic mean of the precision and recall regardless of the category. The experimental results are shown in Table III.

The result shows the proposed algorithm has a 14.16% relative increase in the macro-average F score and a 3% relative increase in the micro-average F score compared to the baseline that only uses review text as features. Thus we can conclude that the vector projection algorithm we proposed to incorporate numeric attributes is effective.

## VI. CONCLUSION

In this paper, we proposed a numeric-attribute-powered sentence embedding algorithm by utilizing a simple vector projection approach. The experimental results demonstrate the effectiveness of this algorithm. Many future research directions are open in this work. For example, we only consider concatenating the text vector representation and the numeric attribute vector representation. Other composition functions can be adopted to learn the information from both sides.

## REFERENCES

[1] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," in *Proc. of the 53rd Annual Meeting of ACL*, 2015.

[2] S. Kuang and B. D. Davison, "Semantic and context-aware linguistic model for bias detection." in *Proc. of the Natural Language Processing meets Journalism IJCAI-16 Workshop*, July 10, 2016, pp. 57–62.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[4] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. of the Conf. on EMNLP*, 2014, pp. 1532–1543.

[5] M. Ma, L. Huang, B. Xiang, and B. Zhou, "Dependency-based convolutional neural networks for sentence embedding," *arXiv preprint arXiv:1507.01839*, 2015.

[6] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in Neural Information Processing Systems*, 2015, pp. 3079–3087.

[7] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. [Online]. Available: http://arxiv.org/abs/1405.4053

[8] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu, "Towards universal paraphrastic sentence embeddings," *CoRR*, vol. abs/1511.08198, 2015.

[9] M. Iyyer, V. Manjunatha, and J. L. Boyd-Graber, "Deep unordered composition rivals syntactic methods for text classification." in *In ACL (1)*, pp. 1681–1691.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. of the Conf. on EMNLP*, vol. 1631, 2013.

[12] Q. Zhao, Y. Shi, and L. Hong, "Gb-cent: Gradient boosted categorical embedding and numerical trees," in *Proc. of the 26th International Conf. on WWW*, 2017, pp. 1311–1319.

[13] S. A. Macskassy, H. Hirsh, A. Banerjee, and A. A. Dayanik, "Using text classifiers for numerical classification," in *Proc. of the 17th IJCAI*, vol. 2. Lawrence Erlbaum Associates, Aug. 2001, pp. 885–890.

[14] A. Madaan, A. Mittal, G. Ramakrishnan, S. Sarawagi *et al.*, "Numerical relation extraction with minimal supervision," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[15] A. T. Chaganty and P. Liang, "How much is 131 million dollars? putting numbers in perspective with compositional descriptions," *arXiv preprint arXiv:1609.00070*, 2016.

[16] C. Gong, D. Tao, J. Yang, and W. Liu, "Teaching-to-learn and learning-to-teach for multi-label propagation." in *AAAI*, 2016, pp. 1610–1616.

[17] T. Kenter, A. Borisov, and M. de Rijke, "Siamese cbow: Optimizing word embeddings for sentence representations," *arXiv preprint arXiv:1606.04640*, 2016.

[18] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," in *Proc. of seventh international conf. on Tools with artificial intelligence, 1995.* IEEE, 1995, pp. 388–391.

[19] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International Conference on Machine Learning*, 2016, pp. 1614–1623.