A Gentle Introduction to the Bag-of-Words Model

by **Jason Brownlee** on October 9, 2017 in **Deep Learning for Natural Language Processing**

Last Updated on August 7, 2019

The bag-of-words model is a way of representing text data when modeling text with machine learning algorithms.

The bag-of-words model is simple to understand and implement and has seen great success in problems such as language modeling and document classification.

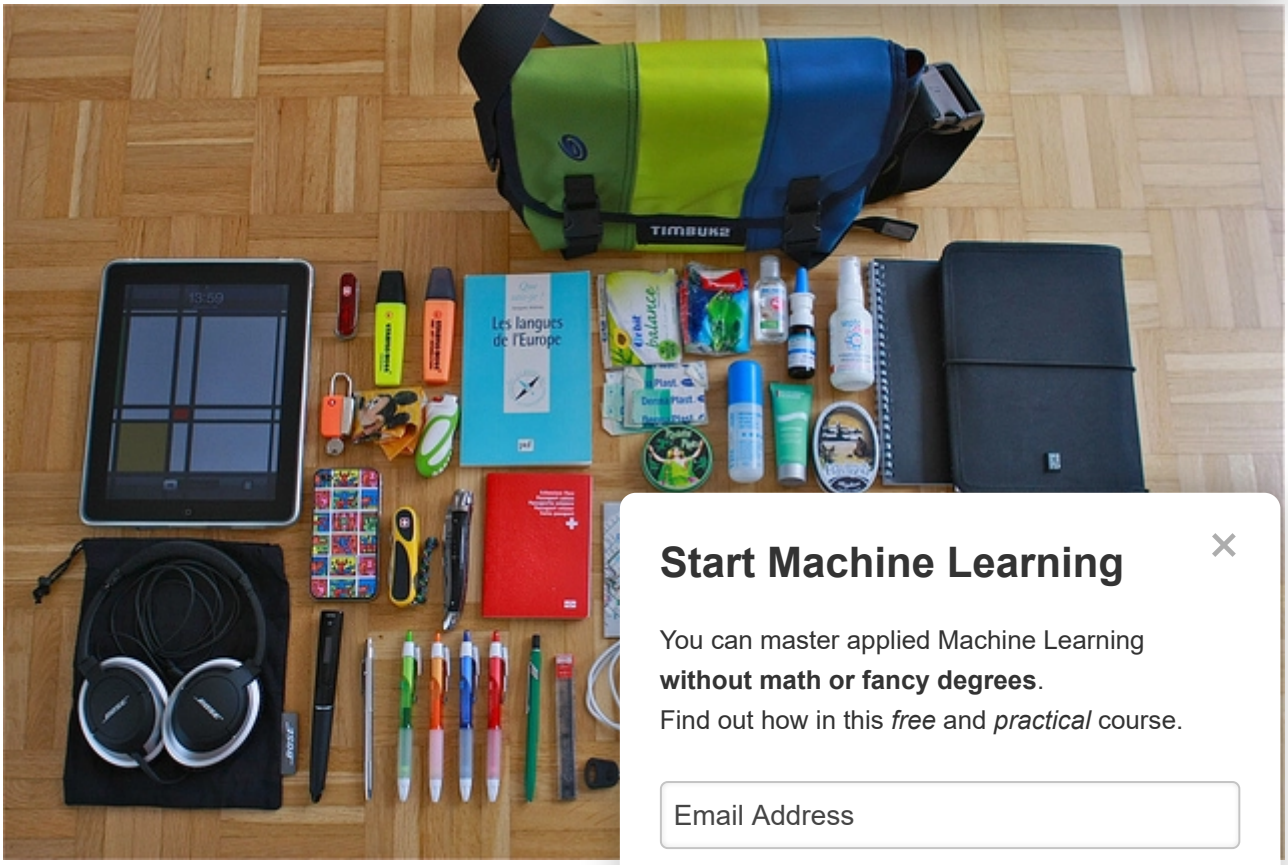
In this tutorial, you will discover the bag-of-words model for feature extraction in natural language processing.

After completing this tutorial, you will know:

- What the bag-of-words model is and why it is needed to represent text.
- How to develop a bag-of-words model for a collection of documents.
- How to use different techniques to prepare a vocabulary and score words.

Kick-start your project with my new book Deep Learning for Natural Language Processing, including *step-by-step tutorials* and the *Python source code* files for all examples.

Let's get started.



A Gentle Introduction t
Photo by Do8y, s

Tutorial Overview

This tutorial is divided into 6 parts; they are:

1. The Problem with Text
2. What is a Bag-of-Words?
3. Example of the Bag-of-Words Model
4. Managing Vocabulary
5. Scoring Words
6. Limitations of Bag-of-Words

The Problem with Text

A problem with modeling text is that it is messy, and techniques like machine learning algorithms prefer well defined fixed-length inputs and outputs.

Machine learning algorithms cannot work with raw text directly; the text must be converted into numbers. Specifically, vectors of numbers.

> *In language processing, the vectors x are derived from textual data, in order to reflect various linguistic properties of the text.*

— Page 65, Neural Network Methods in Natural Language Processing, 2017.

This is called feature extraction or feature encodin

A popular and simple method of feature extraction text.

What is a Bag-of-Words?

A bag-of-words model, or BoW for short, is a way such as with machine learning algorithms.

The approach is very simple and flexible, and can from documents.

A bag-of-words is a representation of text that des involves two things:

1. A vocabulary of known words.
2. A measure of the presence of known words.

It is called a "*bag*" of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

> *A very common feature extraction procedures for sentences and documents is the bag-of-words approach (BOW). In this approach, we look at the histogram of the words within the text, i.e. considering each word count as a feature.*

— Page 69, Neural Network Methods in Natural Language Processing, 2017.

The intuition is that documents are similar if they have similar content. Further, that from the content alone we can learn something about the meaning of the document.

The bag-of-words can be as simple or complex as you like. The complexity comes both in deciding how to design the vocabulary of known words (or tokens) and how to score the presence of known words.

We will take a closer look at both of these concerns

Example of the Bag-of-Words Model

Let's make the bag-of-words model concrete with a worked example.

Step 1: Collect Data

Below is a snippet of the first few lines of text from the book "A Tale of Two Cities" by Charles Dickens, taken from Project Gutenberg.

> *It was the best of times,*
> *it was the worst of times,*
> *it was the age of wisdom,*
> *it was the age of foolishness,*

For this small example, let's treat each line as a se
corpus of documents.

Step 2: Design the Vocabulary

Now we can make a list of all of the words in our m

The unique words here (ignoring case and punctua

- "it"
- "was"
- "the"
- "best"
- "of"
- "times"
- "worst"
- "age"
- "wisdom"
- "foolishness"

That is a vocabulary of 10 words from a corpus containing 24 words.

Step 3: Create Document Vectors

The next step is to score the words in each document.

The objective is to turn each document of free text into a vector that we can use as input or output for a machine learning model.

Because we know the vocabulary has 10 words, we can use a fixed-length document representation of 10, with one position in the vector to score each word.

The simplest scoring method is to mark the presence of words as a boolean value, 0 for absent, 1 for present.

Using the arbitrary ordering of words listed above in our vocabulary, we can step through the first document ("*It was the best of times*") and convert it into a binary vector.

The scoring of the document would look as follows:

- "it" = 1
- "was" = 1
- "the" = 1
- "best" = 1
- "of" = 1
- "times" = 1
- "worst" = 0
- "age" = 0
- "wisdom" = 0
- "foolishness" = 0

As a binary vector, this would look as follows:

```
[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
```

The other three documents would look as follows:

```
"it was the worst of times" = [1, 1, 1, 0,
"it was the age of wisdom" = [1, 1, 1, 0, 1
"it was the age of foolishness" = [1, 1, 1,
```

All ordering of the words is nominally discarded an
from any document in our corpus, ready for use in

New documents that overlap with the vocabulary of known words, but may contain words outside of the vocabulary, can still be encoded, where only the occurrence of known words are scored and unknown words are ignored.

You can see how this might naturally scale to large vocabularies and larger documents.

Managing Vocabulary

As the vocabulary size increases, so does the vector representation of documents.

In the previous example, the length of the document vector is equal to the number of known words.

You can imagine that for a very large corpus, such as thousands of books, that the length of the vector might be thousands or millions of positions. Further, each document may contain very few of the known words in the vocabulary.

This results in a vector with lots of zero scores, called a sparse vector or sparse representation.

Sparse vectors require more memory and computational resources when modeling and the vast number of positions or dimensions can make the modeling process very challenging for traditional algorithms.

As such, there is pressure to decrease the size of the vocabulary when using a bag-of-words model.

There are simple text cleaning techniques that can be used as a first step, such as:

- Ignoring case
- Ignoring punctuation
- Ignoring frequent words that don't contain much information, called stop words, like "a," "of," etc.
- Fixing misspelled words.
- Reducing words to their stem (e.g. "play" from "playing") using stemming algorithms.

A more sophisticated approach is to create a vocabulary of grouped words. This both changes the scope of the vocabulary and allows the bag-of-words to capture a little bit more meaning from the document.

In this approach, each word or token is called a "gr n turn, called a bigram model. Again, only the bigram possible bigrams.

> *An N-gram is an N-token sequence of word a two-word sequence of words like "please tu gram (more commonly called a trigram) is a your", or "turn your homework".*

— Page 85, Speech and Language Processing, 20

For example, the bigrams in the first line of text in t as follows:

- "it was"
- "was the"
- "the best"
- "best of"
- "of times"

A vocabulary then tracks triplets of words is called a trigram model and the general approach is called the n-gram model, where n refers to the number of grouped words.

Often a simple bigram approach is better than a 1-gram bag-of-words model for tasks like documentation classification.

> *a bag-of-bigrams representation is much more powerful than bag-of-words, and in many cases proves very hard to beat.*

— Page 75, Neural Network Methods in Natural Language Processing, 2017.

Scoring Words

Once a vocabulary has been chosen, the occurrence of words in example documents needs to be scored.

In the worked example, we have already seen one very simple approach to scoring: a binary scoring of the presence or absence of words.

Some additional simple scoring methods include:

- **Counts**. Count the number of times each word appears in a document.
- **Frequencies**. Calculate the frequency that each word appears in a document out of all the words in the document.

Word Hashing

You may remember from computer science that a
fixed size set of numbers.

For example, we use them in hash tables when pr
numbers for fast lookup.

We can use a hash representation of known words
having a very large vocabulary for a large text corp
space, which is in turn the size of the vector repres

Words are hashed deterministically to the same int
or count can then be used to score the word.

This is called the "*hash trick*" or "*feature hashing*".

The challenge is to choose a hash space to accommodate the chosen vocabulary size to minimize the probability of collisions and trade-off sparsity.

TF-IDF

A problem with scoring word frequency is that highly frequent words start to dominate in the document (e.g. larger score), but may not contain as much "informational content" to the model as rarer but perhaps domain specific words.

One approach is to rescale the frequency of words by how often they appear in all documents, so that the scores for frequent words like "the" that are also frequent across all documents are penalized.

This approach to scoring is called Term Frequency – Inverse Document Frequency, or TF-IDF for short, where:

- **Term Frequency**: is a scoring of the frequency of the word in the current document.
- **Inverse Document Frequency**: is a scoring of how rare the word is across documents.

The scores are a weighting where not all words are equally as important or interesting.

The scores have the effect of highlighting words that are distinct (contain useful information) in a given document.

> *Thus the idf of a rare term is high, whereas the idf of a frequent term is likely to be low.*

— Page 118, An Introduction to Information Retrieval, 2008.

Limitations of Bag-of-Words

The bag-of-words model is very simple to understand and implement and offers a lot of flexibility for customization on your specific text data.

It has been used with great success on prediction ion classification.

Nevertheless, it suffers from some shortcomings, s

- **Vocabulary**: The vocabulary requires careful ze, which impacts the sparsity of the document re
- **Sparsity**: Sparse representations are harder t nd time complexity) and also for information reas ess so little information in such a large representa
- **Meaning**: Discarding word order ignores the c document (semantics). Context and meaning tell the difference between the same words differe interesting"), synonyms ("old bike" vs "used bi

Further Reading

This section provides more resources on the topic if you are looking go deeper.

Articles

- Bag-of-words model on Wikipedia
- N-gram on Wikipedia
- Feature hashing on Wikipedia
- tf–idf on Wikipedia

Books

- Chapter 6, Neural Network Methods in Natural Language Processing, 2017.
- Chapter 4, Speech and Language Processing, 2009.
- Chapter 6, An Introduction to Information Retrieval, 2008.
- Chapter 6, Foundations of Statistical Natural Language Processing, 1999.

Summary

In this tutorial, you discovered the bag-of-words m

Start Machine Learning

Specifically, you learned:

- What the bag-of-words model is and why we need it.
- How to work through the application of a bag-of-words model to a collection of documents.
- What techniques can be used for preparing a vocabulary and scoring words.

Do you have any questions?
Ask your questions in the comments below and I will do my best to answer.

About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.
View all posts by Jason Brownlee →

‹ How to Develop Word Embeddings in Python with Gensim

What Are Word Embeddings for Text? ›

113 Responses to *A Gentle Introduction to the Bag-of-Words Model*

Samuel October 13, 2017 at 6:00 am #

Great article, thanks for keeping it concise and still easy to understand and read.

REPLY

Jason Brownlee October 13, 2017 at 7:40 am #

Thanks Samuel.

REPLY

meow October 13, 2017 at 4:58 pm #

great read and good references.

REPLY

Jason Brownlee October 14, 2017 at 5:3

Thanks!

Osama Hamed October 18, 2017 at 1:33 am #

It is really a gentle intro.

REPLY

Jason Brownlee October 18, 2017 at 5:39 am #

I hope it helped.

REPLY

Fatma January 9, 2018 at 9:25 pm #

Very helpful and clear step by step explanation.

Thanks.
Fatma

REPLY

Jason Brownlee January 10, 2018 at 5:25 am #

Thanks.

REPLY

Anna January 26, 2018 at 5:55 am #

Hi Jason,

Great article! So, since using Bag-of-Words does not take into account the relation between words or word order. Does transferring the Bag-of-Words model into CNN could tackle the problem and increase the prediction accuracy? I've been searching for the article of implementing BOW + CNN for text classification but no luck so far.

Thank you

REPLY

Jason Brownlee January 27, 2018 at 5:47 am #

No. But you could use a word embedding and an LSTM that would learn the relationship between words.

Nikhil March 22, 2019 at 10:35 pm #

Hi…superb article.

if you have written any articles or any other
understands relationships between words..
please share the links it would be helpful fo

Thank You

Jason Brownlee March 23, 20

Yes, I have many, you can search on the blog or start here:
https://machinelearningmastery.com/start-here/#nlp

REPLY

zenith February 14, 2018 at 4:24 am #

If I understood it correctly, the purpose of word hashing is to easily map the value to the word and get to easily update the count. My question is, would it be easier if I just use a dictionary instead of implementing word hashing?

REPLY

Jason Brownlee February 14, 2018 at 8:25 am #

A dictionary of what?

REPLY

Vince January 17, 2019 at 3:35 am #

Note that in Python, a dictionary IS an
decide what the words map to, though, and I th

maps to its own hashed value. It might be helpful to have a dictionary mapping each word to its own hashed value, if lookups are quicker than your hash function and memory is not a limitation, but you can't really *replace* a hash function with a dictionary.

REPLY

Jason Brownlee January 17, 2019 at 5:29 am #

This would be a set.

REPLY

Georgy March 23, 2018 at 4:14 am #

Thank you for article
i dont actually understand what bag of words is afte
1) The binary vector is the ready BOW model outpu
[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
2) How would it look like if we have more than one
[1, 1, 2, 1, 1, 1, 0, 0, 0, 0] (not a strict example but
3) I dont understand why we cant put bag of words

Jason Brownlee March 23, 2018 at 6:12

The representation of a document as

You can choose how to count, either exists/not-exists, or a count, or something else.

We can plug words into RNNs, often we use a word embedding on the front end to get a more distributed representation of the words:
https://machinelearningmastery.com/what-are-word-embeddings/

Does that help Georgy?

REPLY

Chen-Feng Tsen May 22, 2018 at 5:38 am #

Hello! Thank you for your illustration. We are doing a project of music genre classification based on the song lyrics. However, due to the license issue we only obtained the lyrics in a bag-of-words format and couldn't access the full lyrics. We are trying to use TFIDF, in combination of bag-of-words model. However, in our case we couldn't get document vectors since we don't have information of complete sentences. Do we need to get the full lyric texts to do the training? Or is it sufficient to implement the model with the data we have right now? Thank you very much!

REPLY

Jason Brownlee May 22, 2018 at 6:32 am #

See how far you can get with BoW. To use the embedding/LSTM you will need the original docs.

Lakshmikanth K A June 23, 2018 at 5:19 am #

REPLY

Say I have 10 documents. After removing stop words and stemming etc. I have 50 word vocabulary. My each document would be a vector of 50 tf-idf values which I will model using the dependent variable. That means my modeling data has 10rows*50 features + 1 dependent column..And each cell holds the tf-idf of that vocabulary word. Is this right approach?

Also, tf-if is a value is a function the term and document and all the documents., Since tf comes from what is the term and what is it's frequency in a given document…And idf comes from what is that term's frequency in the overall set of all documents.
Is this understanding right?

Or is tf-idf …After being computed….is summarize

Jason Brownlee June 23, 2018 at 6:21 a

Yes, it is terms described statistically v

zakir January 21, 2020 at 10:03 pm #

Dear Jason I like your Article too

class may contain 2 or 3 comments. Each class comments is different say class 1 has 5 comments and class 2 has 3 comments etc . how i will considered the class as documents and how to convert to CLASS-CLASS metrix

Jason Brownlee January 22, 2020 at 6:23 am #

REPLY

Sounds like a straight multi-class classification problem. You can create a confusion matrix from predictions directly.

Perhaps this will help:
https://machinelearningmastery.com/confusion-matrix-machine-learning/

Nil July 5, 2018 at 4:09 pm #

REPLY

Hi, DR. Jason,

I have two questions, I am seeking for help:
1. I saw something called Term Document Matrix (TDM) in R is it the same thing as Bag-of-Words in Python?
2. I read from one of your posts about Bag-of-Word

having the sparse vector is necessary to convert them in a dense vector before using whit machine learning algorithms.

Best Regards

REPLY

Jason Brownlee July 6, 2018 at 6:39 am #

I don't know about TDM sorry.

No need to convert to dense.

Nil July 7, 2018 at 2:16 am #

Understood. Thanks.

Enrico Marzon July 7, 2018 at 12:07 pm #

Hi. I just want to ask if I can use the Bag C
the tweets from Twitter, then I need to filter those tw
filtered data will be used for classification.

I need your help about this. Thank you in advance.

REPLY

Jason Brownlee July 8, 2018 at 6:15 am #

Sorry, I don't have an example of text filtering, I cannot give you good advice.

REPLY

Mohammad July 14, 2018 at 2:27 am #

Hey Dr. Jason,

thank you so much.
It is really a gentle and great introduction.

REPLY

Jason Brownlee July 14, 2018 at 6:19 am #

Thanks!

Valentina Rodrigues July 26, 2018 at 8:27 p

You have mentioned this:
This results in a vector with lots of zero scores, called a sparse vector or sparse representation.

But in Google's ML Crash Course they have mentioned this:

* A dense representation of this sentence must set an integer for all one million cells, placing a 0 in most of them, and a low integer into a few of them.

* A sparse representation of this sentence stores only those cells symbolizing a word actually in the sentence. So, if the sentence contained only 20 unique words, then the sparse representation for the sentence would store an integer in only 20 cells.

Link: https://developers.google.com/machine-learning/glossary/#sparse_features

Jason Brownlee July 27, 2018 at 5:53 ar

Sure. It is saying we don't save the ze

You can learn more here:
https://machinelearningmastery.com/sparse-ma

Adi August 1, 2018 at 2:49 am #

Hi Jason, excellent article. I'm trying to ca
amazon get placed into one cluster, and tweets wit
topics, A, B, C. My incoming stream of tweets is AE
their respective groups. I'm using Spark Streaming, and the StreamingKMeans model to do this.

How can I vectorize tweets such that those vectors when predicted on by the K-Means model, get placed in the same cluster

REPLY

Jason Brownlee August 1, 2018 at 7:47 am #

Sorry, I don't have examples of working with streaming models.

REPLY

Avinish August 5, 2018 at 9:17 pm #

Hi Jason,

How can you model a system where you have a collection of documents mapped to some labels, and some unlabelled examples.

Document label
D1 —- c1
D2 —- c2
D3 —- c3
.

Start Machine Learning

.
.
.
.
Dk —— c1

Two questions here:-

Q1. The lables I might see in the future might be different from what I have at trainig time and even the corpus might change to some extent.So I have to apply semi-supervised or unsupervised learning to learn online(on the fly) and then do better in later predictions for the seen label, classifying into appropriate class(label).

Q2. If i see a label which I have already seen lets say(c1) and I come across similiar feature vector, I just classify it as 1 and if I see label lets sa
seen before but should have the ability to learn this
Basically classifying into bi-class classification as s
(predicting 1) and not having any parent ticket(0).T
descriptions.

I am struggling to devise an architecture for the pro
me regarding this.

Jason Brownlee August 6, 2018 at 6:27

Sorry, I don't have examples of semi-s

REPLY

Ravi Singh August 7, 2018 at 4:48 pm #

Hi, I followed the tutorial and Now I have a model which I trained using Bag of Word, What I did was converted my text into Sparse Matrix and trained the model. It is giving 95 percent accuracy but now I am unable to predict a simple statement using the model.

This is my code –

I have a data frame with 2 classes labels and body.

```
# using bag of word model for the same
cout_vect = CountVectorizer()
# Convert from object to unicode
final_count = cout_vect.fit_transform(df['body'].values.astype('U'))

#model
# Using a classifier for the bag of word representation
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
X_train, X_test, y_train, y_test = train_test_split(fin
```

```
model = Sequential()
model.add(Dense(264, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(3, activation='softmax'))
# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
y_train = np_utils.to_categorical(y_train, num_classes=3)
y_test = np_utils.to_categorical(y_test, num_classes=3)

model.fit(X_train, y_train, epochs=50, batch_size=
model.evaluate(x=X_test, y=y_test, batch_size=Nc
```

Now I want to predict this statement using my mod

```
x = "Your account balance has been deducted for 4

model.predict(x, batch_size=None, verbose=0, ste
```


Jason Brownlee August 8, 2018 at 6:15

Well done.

To make a prediction you must prepare the inp

REPLY

shubham October 26, 2018 at 9:43 pm #

Hi,
I followed this article.I want to ask how can we extract some difficult words(terminologies) from I different document and store it in vector to make it as the vocabulary for the machine.Will BoW be better solution or should i look for something else.

REPLY

Jason Brownlee October 27, 2018 at 5:59 am #

Not sure I follow.

Bag of words and word2vec are two popular representations for text data in machine learning.

REPLY

Mike November 7, 2018 at 12:33 pm #

Really fantastic article. Excellent clarity. Thanks Jason!

Jason Brownlee November 7, 2018 at 2:47 pm #

REPLY

Thanks Mike, glad it helped.

mustafa December 7, 2018 at 10:09 pm #

REPLY

Thanks for this informative article. I wonder
What is the difference between BOW and TF?
Are these same things?

Jason Brownlee December 8, 2018 at 7:

BOW and TF?

Bag of words and term frequency?

Same generally, although the vector can be fill

Sam January 13, 2019 at 1:38 pm #

Hey, thanks for the article, Jason. Very inf

Jason Brownlee January 14, 2019 at 5:22 am #

REPLY

Thanks, I'm glad it helped.

Agung January 24, 2019 at 2:20 am #

REPLY

thanks for the article, Jason.
i have a question.
if I want to do a classification task with TFIDF vector representation, should that technique representation be carried out on all datasets (training data + test data) first, or done separately, on the training data first then then do the test data?

Jason Brownlee January 24, 2019 at 6:46 am #

REPLY

Good question.

Prepare the vocab and encoding on the training dataset, then apply to train and test.

youri dullens January 25, 2019 at 9:44 pm #

REPLY

Dear Jason,
I'm thinking of writing a thesis about using text from a social media platform(twitter or facebook) to measure the social influence of people(maybe just influencers) on purchase behavior of software licences on mobile apps. Do you think the the Bag-of-Words Model is a good fit, or would you suggest other text analysis models?
If you have any recommendations please!

Thanks in advance,
Youri

Jason Brownlee January 26, 2019 at 6:1

I recommend testing a suite of repres
for your specific prediction problem.

Robert Ling February 13, 2019 at 3:16 am #

Thanks, Jason.

I am a reader from China, and you are a minor cele
those machine learning topics. Thanks for your wo

One of my personal question is how long did it take for you to compose of this piece of article?

Jason Brownlee February 13, 2019 at 8:02 am #

REPLY

Thanks!

I try to write one tutorial per day. Usually, I can write a tutorial in a few hours.

Alleria February 15, 2019 at 1:33 am #

REPLY

Really great article! Thanks for sharing!

Jason Brownlee February 15, 2019 at 8:09 am #

REPLY

Thanks, I'm glad it helped!

Elisio Quintino February 20, 2019 at 9:00 pm #

REPLY

Hi Jason,

First of all, thank you for the material.

Under the session Books, I think both “Chapter 6, An Introduction to Information Retrieval, 2008.” and “Chapter 6, Foundations of Statistical Natural Language Processing, 1999.” are pointing to the latter book, so no reference for the first one.

Best regards, Elisio

Jason Brownlee February 21, 2019 at 7:

Thanks, fixed!

Anjani February 26, 2019 at 10:23 pm #

Nice article about BOW explained well

Jason Brownlee February 27, 2019 at 7:

Thanks.

Alex March 13, 2019 at 3:41 am #

REPLY

NIce article. FYI, the hyperlink on Bag-of-words model on Wikipedia leads to `N-Grams`

Jason Brownlee March 13, 2019 at 7:59 am #

REPLY

Thanks Alex, fixed!

Bindhu April 5, 2019 at 3:51 pm #

REPLY

Hi Jason,

Thanks for this article!!

If i have to predict the ‘impact areas’ of a issue/story, with the below features(textual data):
1. ‘Files modified’ as part of the issue.

Start Machine Learning

2. ‘Component’ that the issue belongs to.
3. Related ‘Test cases’.

Which method can be used here to process this data to feed into a Machine learning model?
[1. Data Pre-processing is done.
2. Supervised test data available]

Can you please suggest something here? It will be a great help!!

REPLY

Jason Brownlee April 6, 2019 at 6:40 am #

I don’t know, sounds like an interest p

Perhaps try a few techniques and also see wh
similar problems?

Carla May 2, 2019 at 1:25 am #

great article. In this web (https://unipython
words/) they have plagiarized it, translating it into S

Jason Brownlee May 2, 2019 at 8:04 am

Thanks for letting me know.

REPLY

Hanna July 4, 2019 at 12:37 pm #

Hi Jason, thanks for your clear explanation. Would like to know how do I cite your article? Would you mind to share any reference to your publication/article so that I can cite your research on this topic.

Thank You

REPLY

Jason Brownlee July 4, 2019 at 2:50 pm #

Sure, this shows you how to cite a post:
https://machinelearningmastery.com/faq/single-faq/how-do-i-reference-or-cite-a-book-or-blog-post

REPLY

Manmohan Singh Bohara July 24, 2019 at 12:19 am #

Thanks for great explanation, Jason.

Jason Brownlee July 24, 2019 at 8:01 am #

REPLY

You're welcome, I'm glad it helped.

Jean July 30, 2019 at 4:48 pm #

REPLY

What about cosine similarity?

Jason Brownlee July 31, 2019 at 6:45 ar

Sorry, I don't have a post on that topic

VIVEK SINGH SISODIYA September 2, 20

can you explain Fuzzy bag-of-word cluste

Jason Brownlee September 2, 2019 at 1

Thanks for the suggestion.

Ahmed M. Shahat October 24, 2019 at 5:21 am #

REPLY

Excellent article, introduces fundamental concepts in a direct and straight forward approach. Thanks Jason, looking forward for more related articles.

Jason Brownlee October 24, 2019 at 5:46 am #

REPLY

Thanks.

Martin October 24, 2019 at 7:10 am #

REPLY

Hi, Jason:

In practice, a document isn't encoded as the 'raw' BoW, like '[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]' in the example, but as a one-hot encoding scheme. So a sentence will be encoded as a matrix (number of words * size of vocabulary), not a vector. Is that right?

Jason Brownlee October 24, 2019 at 2:01 pm #

REPLY

Not in this case.

Think of a bag of words as a one hot encoding for a document (or paragraph).

Anbazhagan Mahadevan November 15, 2019 at 8:23 pm #

REPLY

I had an experience like reading the article in my mother tongue, though I am an Indian.

Superb & Nice explanation.

Jason Brownlee November 16, 2019 at 7

Thanks!

brina April 29, 2020 at 9:53 am #

Hi Jason, thanks for the clear and coincide

Would you consider BoW a “word embedding” met

It is my understanding that the BoW method is part of the so-called “vector semantic” and as such it is a form of embedding (i.e. representing) the meaning of a word in a vector. However, I frequently hear people contrasting “BoW” with “word embedding” approaches (and they refer to CBOW or skip-gram for example). This makes me wonder if it is correct defining BoW as a word embedding method.

Thank you in advance for your answer!

Jason Brownlee April 29, 2020 at 12:06 pm #

REPLY

You’re welcome.

Maybe. Not really, there is no relationship between words reflected in the representation. It is distributed though.

Jennifer May 18, 2020 at 5:16 pm #

REPLY

The main disadvantage is that it does not take account of word order, so it looses important aspects of meaning ,It can’t take account of similarity between different words (word embeddings is a solution to this), It is a large representation that inc

which will be zero for a given text..

Jason Brownlee May 19, 2020 at 5:57 am #

REPLY

Agreed.

hadi June 1, 2020 at 10:43 pm #

REPLY

great explanation and article

i just have one question

if i am using bag of words in sentiment analysis to
and any machine learning classifier used with bag
know the total tweet positive or negative
according to what the classifier know this tweet pos
every word in the tweet have one vector from the b
all the vectors of all words in the same tweet to spe

Jason Brownlee June 2, 2020 at 6:13 am

How does the model know it is positiv
data.

Perhaps I don't understand your question, if so, perhaps you can restate it.

hadi June 2, 2020 at 8:55 am #

REPLY

thanks for reply

i mean how the machine learning classifier identify the polarity of tweet with only bag of word model

we dont use any rules or lexicon to extract sentiment words from the tweet then apply any rule on this sentiment (aggregation or any other rule) to say that all this tweet is positive or negative

we only have all the words and its count how this work

Jason Brownlee June 2, 2020 at 1:19 pm #

REPLY

It learns the relationship between words and the target class label. It solves the problem because we cannot code the solution explicitly.

hadi June 3, 2020 at 12:48 am #

thanks for reply all the time

any resources to understand the details deep

REPLY

Jason Brownlee June 3, 2020 at 8:01 am #

Yes see the resources listed in the “Further Reading” section.

REPLY

hadi June 4, 2020 at 3:18 am #

thank you or help

Jason Brownlee June 4, 2020 at 6:26 am

You’re welcome!

Parul June 9, 2020 at 3:28 pm #

Awesome article Jason ! This explained w

REPLY

Jason Brownlee June 10, 2020 at 6:07 am #

Thanks!

REPLY

Abdelrahman July 7, 2020 at 11:29 pm #

Thanks for your simplicity to deliver the information.

REPLY

Jason Brownlee July 8, 2020 at 6:32 am #

You’re welcome.

REPLY

Utsav Rastogi July 12, 2020 at 2:51 pm #

This is one of the best articles I’ve ever read in this field. Great Work Jason! will follow this website frequently to clear my doubts.

Jason Brownlee July 13, 2020 at 5:55 am #

REPLY

Thank you, I'm happy to hear that.

Paul Joseph October 12, 2020 at 3:59 am #

REPLY

Best explanation ever seen.
Surfed through many sites, but not satisfied.
Thanks to Mr.Jason for crisp,clear and precise exp

Jason Brownlee October 12, 2020 at 6:4

Thanks!

Indika October 14, 2020 at 2:29 am #

Could you please mention steps for identi
python libry, I followed your tutorial on this but it on
what i do next. I can search the internet if i know ea
Thank you for your great tutorials.

Jason Brownlee October 14, 2020 at 6:24 am #

REPLY

Sorry, I don't have a tutorial on coding BoW models from scratch.

If you decide to use a library see this:
https://machinelearningmastery.com/?s=movie+review&post_type=post&submit=Search

Vaishali October 17, 2020 at 5:36 am #

REPLY

Thanks a lot. Very good article, Clear BOW very well.

Jason Brownlee October 17, 2020 at 6:13 am #

REPLY

You're welcome.

sena mosisa January 4, 2021 at 5:37 pm #

REPLY

thank for your sharing this iedea i went to know how to convert categorical data to numerical data using one hot encoding i went to know this please help me

Jason Brownlee January 5, 2021 at 6:16 am #

REPLY

Here is an example:
https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/

osama January 30, 2021 at 11:58 pm #

can we consider the bag of words 3 types
frequencies and if this is true can we consider tfidf
bag of words

Jason Brownlee January 31, 2021 at 5:3

Yes, TF/IDF is like an advanced bag c

osama February 3, 2021 at 11:40 pm #

REPLY

thank you for reply and help all the time my question is if i want to say
To convert our cleaned review to numerical feature vectors we can do use the following methods
:
1- bag of words
2- tfidf
3-word2vec or glove
this is correct or i must say the methods is
1- bag of words
2-word2vec or glove
and in this case i consider tfidf inside the bag of words
thanks in advance

Jason Brownlee February 4, 2021 at 6:20 am #

REPLY

Yes, that is a good start.

Start Machine Learning

osama February 4, 2021 at 11:51 am #

you mean this is correct if i say it
1- bag of words
2- tfidf
3-word2vec or glove

Jason Brownlee February 4, 2021 at 1:39 pm #

Yes, you can use those m

osama February 6, 2021 at 1:04 am #

thank you for help all the time and your ni

Jason Brownlee February 6, 2021 at 5:5

You're welcome!

Leave a Reply

Name (required)

Email (will not be published) (required)

Website

SUBMIT COMMENT

Welcome!
I'm *Jason Brownlee* PhD
and I **help developers** get results with **machine learning**.
Read more

Never miss a tutorial:

Picked for you:


How to Develop a Deep Learning Photo Captior


How to Develop a Neural Machine Translation S


How to Use Word Embedding Layers for Deep l


How to Develop a Word-Level Neural Language


How to Develop a Seq2Seq Model for Neural Machine Translation in Keras

Loving the Tutorials?

The Deep Learning for NLP EBook is
where you'll find the ***Really Good*** stuff.

>> SEE WHAT'S INSIDE

Start Machine Learning