Title: Bag-of-words model

URL: https://en.wikipedia.org/wiki/Bag-of-words_model

PageID: 14003441

Categories: Category:Machine learning, Category:Natural language processing

---

The bag-of-words (BoW) model is a model of text which uses an unordered collection (a " bag ") of words. It is used in natural language processing and information retrieval (IR). It disregards word order (and thus most of syntax or grammar) but captures multiplicity .

The bag-of-words model is commonly used in methods of document classification where, for example, the (frequency of) occurrence of each word is used as a feature for training a classifier . It has also been used for computer vision .

An early reference to "bag of words" in a linguistic context can be found in Zellig Harris 's 1954 article on Distributional Structure .

Definition

The following models a text document using bag-of-words. Here are two simple text documents:

Based on these two text documents, a list is constructed as follows for each document:

Representing each bag-of-words as a JSON object , and attributing to the respective JavaScript variable:

Each key is the word, and each value is the number of occurrences of that word in the given text document.

The order of elements is free, so, for example {"too":1,"Mary":1,"movies":2,"John":1,"watch":1,"likes":2,"to":1} is also equivalent to BoW1 . It is also what we expect from a strict JSON object representation.

Note: if another document is like a union of these two,

its JavaScript representation will be:

So, as we see in the bag algebra , the "union" of two documents in the bags-of-words representation is, formally, the disjoint union , summing the multiplicities of each element.

Word order

The BoW representation of a text removes all word ordering. For example, the BoW representation of " man bites dog " and "dog bites man" are the same, so any algorithm that operates with a BoW representation of text must treat them in the same way. Despite this lack of syntax or grammar, BoW representation is fast and may be sufficient for simple tasks that do not require word order. For instance, for document classification , if the words "stocks" "trade" "investors" appears multiple times, then the text is likely a financial report, even though it would be insufficient to distinguish between

Yesterday, investors were rallying, but today, they are retreating.

and

Yesterday, investors were retreating, but today, they are rallying.

and so the BoW representation would be insufficient to determine the detailed meaning of the document.

Implementations

Implementations of the bag-of-words model might involve using frequencies of words in a document to represent its contents. The frequencies can be "normalized" by the inverse of document frequency, or tf–idf . Additionally, for the specific purpose of classification, supervised alternatives have been developed to account for the class label of a document. Lastly, binary (presence/absence or 1/0) weighting is used in place of frequencies for some problems (e.g., this option is implemented in the WEKA machine learning software system).

Python implementation

Hashing trick

A common alternative to using dictionaries is the hashing trick , where words are mapped directly to indices with a hashing function. Thus, no memory is required to store a dictionary. Hash collisions are typically dealt via freed-up memory to increase the number of hash buckets [clarification needed] . In practice, hashing simplifies the implementation of bag-of-words models and improves scalability.

See also

Additive smoothing

Feature extraction

Machine learning

MinHash

Vector space model

w-shingling

Notes

References

McTear, Michael (et al) (2016). The Conversational Interface . Springer International Publishing.

v

t

e

AI-complete

Bag-of-words

n -gram Bigram Trigram

Bigram

Trigram

Computational linguistics

Natural language understanding

Stop words

Text processing

Argument mining

Collocation extraction

Concept mining

Coreference resolution

Deep linguistic processing

Distant reading

Information extraction
Named-entity recognition
Ontology learning
Parsing Semantic parsing Syntactic parsing
Semantic parsing
Syntactic parsing
Part-of-speech tagging
Semantic analysis
Semantic role labeling
Semantic decomposition
Semantic similarity
Sentiment analysis
Terminology extraction
Text mining
Textual entailment
Truecasing
Word-sense disambiguation
Word-sense induction
Compound-term processing
Lemmatisation
Lexical analysis
Text chunking
Stemming
Sentence segmentation
Word segmentation
Multi-document summarization
Sentence extraction
Text simplification
Computer-assisted
Example-based
Rule-based
Statistical
Transfer-based
Neural
BERT
Document-term matrix
Explicit semantic analysis
fastText

GloVe
Language model (large)
Latent semantic analysis
Seq2seq
Word embedding
Word2vec
Corpus linguistics
Lexical resource
Linguistic Linked Open Data
Machine-readable dictionary
Parallel text
PropBank
Semantic network
Simple Knowledge Organization System
Speech corpus
Text corpus
Thesaurus (information retrieval)
Treebank
Universal Dependencies
BabelNet
Bank of English
DBpedia
FrameNet
Google Ngram Viewer
UBY
WordNet
Wikidata
Speech recognition
Speech segmentation
Speech synthesis
Natural language generation
Optical character recognition
Document classification
Latent Dirichlet allocation
Pachinko allocation
Automated essay scoring
Concordancer
Grammar checker

Predictive text

Pronunciation assessment

Spell checker

Chatbot

Interactive fiction

Question answering

Virtual assistant

Voice user interface

Formal semantics

Hallucination

Natural Language Toolkit

spaCy