Bag-of-Words Technique in Natural Language Processing: A Primer for Radiologists

Krishna Juluru, MD
Hao-Hsin Shih, MS
Krishna Nand Keshava Murthy, MS
Pierre Elnajjar, MS

Abbreviations: BOW = bag of words, IDF = inverse document frequency, ML = machine learning, NLP = natural language processing, NLTK = Natural Language Toolkit, TF = term frequency, TFIDF = term frequency–inverse document frequency

RadioGraphics 2021; 41:1420–1426

https://doi.org/10.1148/rg.2021210025

Content Codes: AI IN

From the Department of Radiology, Memorial Sloan Kettering Cancer Center, 1275 York Ave, Box 29, New York, NY 10065. Recipient of a Certificate of Merit award for an education exhibit at the 2020 RSNA Annual Meeting. Received February 7, 2021; revision requested March 19 and received April 2; accepted April 2. For this journal-based SA-CME activity, the author K.J. has provided disclosures (see end of article); all other authors, the editor, and the reviewers have disclosed no relevant relationships. **Address correspondence to** K.J.L. (e-mail: *juluruk@mskcc.org*).

Supported in part by the National Institutes of Health/National Cancer Institute Cancer Center Support Grant (P30 CA008748).

SA-CME LEARNING OBJECTIVES

After completing this journal-based SA-CME activity, participants will be able to:

- Describe several preprocessing steps in preparing text for feature extraction.
- Understand how BOW technique represents free text as features.
- Discuss techniques for providing different values to words in a corpus of documents.

See www.rsna.org/education/search/RG.

Natural language processing (NLP) is a methodology designed to extract concepts and meaning from human-generated unstructured (free-form) text. It is intended to be implemented by using computer algorithms so that it can be run on a corpus of documents quickly and reliably. To enable machine learning (ML) techniques in NLP, free-form text must be converted to a numerical representation. After several stages of preprocessing including tokenization, removal of stop words, token normalization, and creation of a master dictionary, the bag-of-words (BOW) technique can be used to represent each remaining word as a feature of the document. The preprocessing steps simplify the documents but also potentially degrade meaning. The values of the features in BOW can be modified by using techniques such as term count, term frequency, and term frequency–inverse document frequency. Experience and experimentation will guide decisions on which specific techniques will optimize ML performance. These and other NLP techniques are being applied in radiology. Radiologists' understanding of the strengths and limitations of these techniques will help in communication with data scientists and in implementation for specific tasks.

Online supplemental material is available for this article.

Introduction

Natural language processing (NLP) is a methodology designed to extract concepts and meaning from human-generated unstructured (free-form) text. It is intended to be implemented by using computer algorithms so that it can be run on a large volume of documents quickly and reliably. Cai et al (1) have described the key approaches and technologies in NLP. In one approach, concepts are first extracted from free-form text by using pattern-matching and linguistic analysis techniques. Human expert–developed rules are then used to translate the concepts into meaning. For example, a chest radiography report describing "new right lower lobe opacity" can be processed into concepts of temporality (new), anatomic location (right lung lower lobe), and lesion description (opacity). An expert may then devise a rule that states that this combination of concepts infers a meaning of pneumonia. While these rules are easy to understand, they can be increasingly cumbersome to generate as the complexity of concepts and number of combinations grow.

TEACHING POINTS

- In NLP and linguistics, the process of breaking text into individual meaningful components is known as tokenization. One form of tokenization is the identification of individual words in a sentence.
- Stop words in NLP are words that occur with the highest frequency in speech and writing within the target language. They are commonly used when constructing sentences and typically do not carry any special meaning (eg, "the," "is").
- The NLTK library includes a list of 127 stop words, including "from," "with," "for," and "or." Other words include, notably, "no," "not," and "if," which when removed could drastically change meaning, and therefore caution should be exercised when removing stop words.
- Token normalization in NLP is the process of reducing words to a root form so that variations of the same word are recognized as a single entity. One method is known as stemming, a rules-based approach in which markers for shades of meaning such as plurality and tense are modified or removed.
- If a word appears in every document of the corpus, then that word does not help to identify the unique aspects of a given document. It is the unique aspects of a document that are essential to ML prediction models.

Statistical and machine learning (ML) methods can be used to improve rules-based approaches or to replace them entirely. Implementation requires a set of features and training data to create models. Once trained, the models accept previously unseen features as an input to make predictions.

In this article, we review a popular method of feature extraction known as the bag-of-words (BOW) technique to familiarize radiologists with this approach in NLP and help improve their communication with data scientists with whom they work.

Text Preprocessing

ML algorithms do not accept free text directly. Instead, they require as an input a set of properties about the entity being analyzed, also known as features. The presence or absence of features is then stored into a computational entity known as a vector. We define *ground truth* as a predetermined known answer. Multiple feature vectors, along with ground truth labels, are the inputs to an ML algorithm during training. Once trained, the ML algorithm is able to make predictions when analyzing a new set of feature vectors.

For example, when describing a fruit, we can decide to observe the following features: color and shape. In a basket containing apples and bananas, we conclude that there exist two colors (red and yellow) and two shapes (round and curvilinear), and we construct a feature vector with these four features (Table 1). As we assess the individual fruits in the basket, we ask, "Is the item red?" If so, we place a "1" into the corresponding cell in the feature vector, and if not, we place a "0" in that cell. For the same fruit, moving to another location in the feature vector, we ask, "Is the item round?" If so, we place a "1" in the corresponding cell in that feature vector, and if not, we place a "0" in that cell. We repeat this process for all features.

We then define a ground truth–label vector containing two possibilities, "apple" and "banana." During the training phase of an ML algorithm, we must provide the algorithm a ground truth label. For an apple, we place a "1" in the "apple" cell of the ground truth vector and a "0" in the "banana" cell. A future trained ML algorithm will take as an input a vector containing four features and output a vector containing the probabilities that the input vector describes the two possible ground truths, "apple" or "banana."

In the fruit example, we characterized the two fruits by a set of numbers stored in feature vectors. The BOW technique accomplishes the same for free text. A collection of free text is defined to be a *document* and a collection of documents is defined to be a *corpus*. The features are the words themselves, and a list of all possible unique words in the corpus is known as a dictionary. In preparing a corpus to be presented in a BOW representation, a data scientist will employ some basic strategies, including *(a)* tokenization, *(b)* removing stop words, *(c)* token normalization, and *(d)* creating a master dictionary.

Many of these steps can be automated by using prebuilt collections of programs known in computer science as libraries. The Natural Language Toolkit (NLTK) is a library that performs a variety of NLP functions and is written in the Python (Python Software Foundation, *https://www.python.org*) programming language (2,3). To demonstrate some basic strategies in text preparation for BOW, we will work through an example challenge in which our goal is to devise an ML-based decision-support tool.

Challenge: Prepare Text to Create a Decision-Support Tool to Predict Appropriate Radiologic Examinations

Toward building a BOW representation of free text, we will work through a use case in which the end goal is to create a decision-support tool that will help physicians identify the appropriate radiologic examinations to order from the clinical statements (or history) provided. Therefore, our inputs are the clinical statements provided for any radiologic examination. The outputs are the appropriate examination to order. In the training phase, we need to provide the ML algorithm with both the inputs and the ground truth–label outputs. When fully trained, we hope the ML

Table 1: Fruit Features and Ground Truth Vectors

	Features Vector				Ground Truth Vector	
Fruit Number	Color: red	Color: yellow	Shape: round	Shape: curvilinear	Apple	Banana
1	1	0	1	0	1	0
2	0	1	0	1	0	1

Table 2: Clinical Statements before and after Text Preprocessing

Document Number	Clinical Statement	Appropriate Radiologic Examination (Ground Truth)
Before text preprocessing		
1	Pancreatic cancer with metastasis. Jaundice with transaminitis, evaluate for obstruction process.	CT of the abdomen with contrast material
2	Pancreatitis. Breast cancer. No output from enteric tube. Assess tube.	Chest radiography, one view
3	Metastatic pancreatic cancer. Acute renal failure, evaluate for hydronephrosis or obstructive uropathy.	Renal US
After text preprocessing		
1	pancreat cancer metastasi jaundic transamin evalu obstruct process	CT of the abdomen with contrast material
2	pancreat breast cancer ouput enter tube assess tube	Chest radiography, one view
3	metastat pancreat cancer acut renal failur evalu hydronephrosi obstruct uropathi	Renal US

model will predict the appropriate examination to order from the inputs alone. To obtain our starting point data, we run a query in our electronic medical records system for all radiologic examinations ordered in a certain time period, with specific requests to obtain the clinical statements provided and examinations performed. Depending on the time period set, such a query can yield a few to hundreds of thousands of results. For the purposes of simplicity and clarity, we will work with only three results in this example. Therefore, our corpus consists of three documents, in which each document is a clinical statement (Table 2). A hands-on exercise is available as a companion to this article on Github (*https://github.com/RSNA/RadioGraphics/blob/main/Bag%20of%20Words%20Technique%20in%20Natural%20Language%20Processing:%20A%20Primer%20for%20Radiologists.ipynb*).

Tokenization

In NLP and linguistics, the process of breaking text into individual meaningful components is known as tokenization (4). One form of tokenization is the identification of individual words in a sentence. In a BOW representation, the features of a document are the individual words and can be identified simply by a form of pattern matching, in which each word is defined as a string of nonspace characters and where a new word begins after a blank space. In an application such as Microsoft Excel (Microsoft), text can be broken into individual words by using the "Text to Columns" tool and setting the delimiter to a blank space. By using NLTK in Python, we can use a function known as "tokenizer," which when applied to our corpus yields the following results:

- Doc 1: ['Pancreatic', 'cancer', 'with', 'metastasis', 'Jaundice', 'with', 'transaminitis', 'evaluate', 'for', 'obstruction', 'process']
- Doc 2: ['Pancreatitis', 'Breast', 'cancer', 'No', 'output', 'from', 'enteric', 'tube', 'Assess', 'tube']
- Doc 3: ['Metastatic', 'pancreatic', 'cancer', 'Acute', 'renal', 'failure', 'evaluate', 'for', 'hydronephrosis', 'or', 'obstructive', 'uropathy'].

In this representation, in which "Doc" represents document, each word exists on its own, without the context provided by its neighbors. Words remain associated by their existence in the same document and by no other means. For example, in the first document, "jaundice" is no longer associated with "transaminitis," and "evaluate" is no longer associated with "obstructive process." Therefore, some meaning has already been lost. From this step, we identify 33

total words and 26 unique words. These unique words came from a corpus of three small documents. With larger documents containing more words and a larger corpus, the number of unique words grows rapidly and can reduce the speed and performance of ML algorithms. We therefore seek ways of limiting the unique words by focusing on the ones that are essential.

Removing Stop Words

Stop words in NLP are words that occur with the highest frequency in speech and writing within the target language (5). They are commonly used when constructing sentences and typically do not carry any special meaning (eg, "the," "is"). Their removal will greatly reduce the size of the dictionary of unique words, a desirable result. There is no single list of universally accepted stop words, and variations exist in different stop word dictionaries. The NLTK library includes a list of 127 stop words, including "from," "with," "for," and "or" (Table E1). Other words include, notably, "no," "not," and "if," which when removed could drastically change meaning, and therefore caution should be exercised when removing stop words. In our example, removing all stop words found in the NLTK library from our current documents yields the following result:

- Doc 1: ['Pancreatic', 'cancer', 'metastasis', 'Jaundice', 'transaminitis', 'evaluate', 'obstruction', 'process']
- Doc 2: ['Pancreatitis', 'Breast', 'cancer', 'output', 'enteric', 'tube', 'Assess', 'tube']
- Doc 3: ['Metastatic', 'pancreatic', 'cancer', 'Acute', 'renal', 'failure', 'evaluate', 'hydronephrosis', 'obstructive', 'uropathy'].

The word "no" from the second document has been removed, along with several other stop words. Again, some meaning has been lost, and this could change the functionality of a downstream application. We now identify 26 total words and 21 unique words. However, while words such as "obstruction" and "obstructive" are unique in one sense, they do refer to the same concept of blockage. The same is true for the words "metastasis" and "metastatic." We therefore desire a way to treat multiple forms of a word as the same entity.

Token Normalization

Token normalization in NLP is the process of reducing words to a root form so that variations of the same word are recognized as a single entity (6). One method is known as stemming, a rules-based approach in which markers for shades of meaning such as plurality and tense are modified or removed (7). The stem may be a word itself but need not be so. One of the most widely used stemming algorithms is known as the Porter stemming algorithm, which is also available in the NLTK library and can be run on a large corpus with a few lines of code (8). Applying this algorithm to the words "grows," "leaves," and "argued," for example, results in the stems "grow," "leav," and "argu," respectively. Although the stems "leav" and "argu" are no longer proper words in the English language, the use of a common stem for multiple forms of the same word simplifies downstream NLP techniques.

Another method of token normalization known as lemmatization uses a dictionary of known word variations instead of a rules-based approach (9). The NLTK library also includes lemmatizer algorithms that can quickly process a corpus of documents. Applying the lemmatizer to the words "grows," "leaves," and "argued," for example, results in the words "grow," "leaf," and "argued." While the word "grows" was reduced to "grow" in both methods, we see some important differences with the other words. The word "leaves" was reduced to "leaf" by the lemmatizer. If the intended meaning of "leaves" was parts of a tree, then this reduction is acceptable. However, if the intended meaning was that which could be used in the sentence "Johnny leaves quickly," then the lemmatizer has changed meaning.

For the purposes of building our BOW model, we will proceed with stemming alone. We continue to describe the stems of words as "words" in this article for simplicity. Applying the Porter stemming algorithm in NLTK to our documents yields the following results:

- Doc 1: ['pancreat', 'cancer', 'metastasi', 'jaundic', 'transamin', 'evalu', 'obstruct', 'process']
- Doc 2: ['pancreat', 'breast', 'cancer', 'output', 'enter', 'tube', 'assess', 'tube']
- Doc 3: ['metastat', 'pancreat', 'cancer', 'acut', 'renal', 'failur', 'evalu', 'hydronephrosi', 'obstruct', 'uropathi']

As we desired, the words "obstruction" and "obstructive" have been reduced to a single stem, "obstruct." However, the words "metastasis" and "metastatic" were reduced to the stems "metastasi" and "metastat" and therefore were not unified into a single stem, although they have the same underlying meaning. Notably, the words "pancreatic" and "pancreatitis" were reduced to a single stem, "pancreat." A potentially important concept of inflammation of the pancreas as expressed in the word "pancreatitis" has been removed. This could be an undesirable result, depending on the downstream application of the NLP. Other words such as "cancer" and "breast"

were unchanged. After stemming, we identified 26 total words and 19 unique words. The clinical statements following preprocessing are shown in Table 2.

Creating a Master Dictionary

A *dictionary* is defined as a list of allowable terms. In our fruit example, our dictionary of features includes four items. In building our BOW model in the current example, our dictionary of features is the 19 unique words available after tokenization, removal of stop words, and stemming from the corpus of three documents. A simple Python function known as "set" can take as an input a list of any number of words and output the unique words in that list. The individual words lack context, associated modifiers such as "no" may have been removed, and, as we have seen, the meaning of the words may have been changed in the preprocessing steps.

BOW Representations and Measuring the Value of Words

With a master dictionary of features, we can now perform feature extraction from individual documents into a feature vector. In the same way that we assessed each fruit by asking "Is this fruit red?" or "Is this fruit round?", we assess each document from our corpus and ask, "Does this document contain the word 'acut'?" If so, we place some value in the corresponding cell of the feature vector. In the simplest approach, the value placed is based on term presence, where we place a "0" or "1" when a given term (feature) is absent or present, respectively. There are several other ways to assess the value of words and thereby steer ML algorithms to weight some words more heavily than others. These include term count, term frequency (TF), and term frequency–inverse document frequency (TFIDF).

Term Presence

Our first BOW representation for the three example documents is shown in Table E2. This is the simplest representation of the feature vectors, in which the features are given a binary value of "0" or "1" depending on absence or presence in a given document (10–12).

Term Count

Term count replaces the binary representation of a feature with a count of the number of times a term is present in the document. The assumption here is that the more times a word appears in a document, the more important it is (10–12). Table E3 is a representation of term count, and we see that the cell corresponding to document 2 and the feature "tube" shows "2" because this word appears twice in that document.

Term Frequency

TF is defined as the ratio of the number of times a term appears in a document to the number of terms in the document (10–15). TF therefore normalizes the importance given to a word with respect to the size of the document. Like term count, TF places a higher value on words that appear more often in a document, but as the size of a document increases, the TF for any given term will decrease. Table E4 is a representation of TF. We perform the calculation of the TF for the feature "obstruct" as an example.

In document 1, there are a total of eight terms, and the term "obstruct" occurs one time. In document 3, there are a total of 10 terms, and the term "obstruct" also occurs one time. The TF for the term "obstruct" in document 1 is one per eight, or 0.125. The TF for the term "obstruct" in document 3 is 1/10, or 0.1. The term count for "obstruct" is assigned the same value for documents 1 and 3. However, the TF for "obstruct" is greater in document 1 than document 3. Therefore, when provided with feature vectors using TF, an ML algorithm will place more value to the word "obstruct" in document 1 than it will in document 3.

Term Frequency–Inverse Document Frequency

Another consideration in assessing the value of a particular word is the number of times a given word appears in the overall corpus. If a word appears in every document of the corpus, then that word does not help to identify the unique aspects of a given document. It is the unique aspects of a document that are essential to ML prediction models. Inverse document frequency (IDF) is a method used to increase the value of rare words and decrease the value of common words in the corpus (10–13,15):

$$IDF = log\frac{\#\ documents\ in\ corpus}{\#\ documents\ where\ term\ appears}$$

in which *log* represents logarithm.

Unlike the previously described metrics, the IDF for any given term applies to the entire corpus, not to any individual document. The IDF for the terms in our example is shown in Table E5. To use the IDF to weight the feature vector in any given document, we multiply it with its TF to obtain a new metric simply described as TFIDF (Table E6). We perform the calculation of the TFIDF for the feature "cancer" as an example. The number of documents in our example corpus is three. The number of documents in which

the term "cancer" appears is also three. The IDF for "cancer" is therefore log 3 per 3, or zero. The TFs for the term "cancer" in documents 1, 2, and 3 are 0.12, 0.12, and 0.1, respectively. However, the TFIDF for the term "cancer" is zero for all documents. Therefore, when provided feature vectors using TFIDF from our example corpus, an ML model will place no value on the word "cancer." This is intuitively understandable. If "cancer" appears in all the documents, then the word cannot be useful in differentiating the documents and predicting our desired outcome of the appropriate radiologic examination.

Next Steps

We have defined several numeric representations of free text, known as feature vectors, in three documents. The simplest BOW technique values individual words by their presence or absence only. More sophisticated variations of this technique value words by how often they appear in individual documents and in the corpus. During the training phase, an ML model needs to be provided with the feature vectors of each document, as well as the ground truth labels. In our example, the prediction desired is one of the three radiologic examinations. The ground truth representation also takes the form of a feature vector, containing three possible labels, with a "1" placed in the cell corresponding to the appropriate examination for a given document (Table E7). For any given document, the feature vector and the corresponding ground truth vector are provided to the ML algorithm for training. The choice of the feature vector to use for training and the choice of the ML model are decisions of experimentation and experience, along with a multitude of parameters that can be optimized. A variety of techniques have been described in radiology literature (16–18). The goal is to find the appropriate combination that maximizes prediction performance (19) on an independent test set.

Limitations

NLP, in a broad sense, aims to derive meaning from a variety of input text. While the simplest forms of BOW models have been applied effectively in a variety of applications, there are limitations. Preprocessing steps aimed at reducing complexity and variability can adversely alter meaning. We observed this happening when removing stop words that involve negation and when using lemmitizers ("leaves" changing to "leaf"). Furthermore, because the BOW model in its simplest form considers each word in a document individually, the meaning derived from words that are spatially associated with one another is lost. For example, a simple BOW representation of the statement "Fever. Urinary catheter with no output," would remove the association of "no" with "output," leaving the possibility that "no" is associated with other words such as "fever." To address this problem, variations of the BOW technique include dictionaries with terms containing multiple words, known as *n*-grams. A 2-gram representation of the statement above would include the terms "fever urinary," "urinary catheter," "catheter with," "with no," and "no output." While *n*-grams may help preserve meaning in individual documents, they may make harder the effort of finding similar meanings within multiple documents.

Conclusion

BOW is a method used in NLP to convert free text to numeric representation in preparation for ML. A variety of preprocessing steps can be used to reduce the size and variability of free text, but they can also potentially alter the meaning of some documents. Several different techniques can be used to assign different weights to words in documents, which can potentially affect the performance of a trained algorithm. Radiologists interested in applying NLP to their practice should be familiar with the strengths and limitations of these techniques so they can better communicate with data scientists to develop effective solutions.

Acknowledgments.—We thank Fen Yik, PhD, for reviewing the linguistic portions of this manuscript. We thank Joanne Chin, MFA, for valuable assistance in revising and preparing this manuscript.

Disclosures of Conflicts of Interest.—**K.J.** *Activities related to the present article:* editorial board member of *RadioGraphics* (not involved in the handling of this article). *Activities not related to the present article:* disclosed no relevant relationships. *Other activities:* disclosed no relevant relationships.

References

1. Cai T, Giannopoulos AA, Yu S, et al. Natural language processing technologies in radiology research and clinical applications. RadioGraphics 2016;36(1):176–191.
2. Bird S, Klein E, Loper E. Natural Language Processing with Python : Analyzing Text with the Natural Language Toolkit. http://www.nltk.org/book/. Accessed January 24, 2021.
3. Natural Language Toolkit: NLTK 3.5 documentation. https://www.nltk.org. Last update April 20, 2021. Accessed January 24, 2021.
4. Grefenstette G. Tokenization. In: van Halteren H, ed. Syntactic Wordclass Tagging: Text, Speech and Language Technology, vol 9. Dordrecht, the Netherlands: Springer, 1999; 117–133.
5. Gerlach M, Shi H, Amaral LAN. A universal information theoretic approach to the identification of stopwords. Nat Mach Intell 2019;1(12):606–612.
6. Fautsch C, Savoy J. Algorithmic stemmers or morphological analysis? An evaluation. J Am Soc Inf Sci 2009;60(8):1616–1624.
7. Singh J, Gupta V. A systematic review of text stemming techniques. Artif Intell Rev 2017;48(2):157–217.

8. Willett P. The Porter stemming algorithm: then and now. Program 2006;40(3):219–223.
9. Ting M, Kadir RA, Sembok TMT, Ahmad F, Azman A. Adaptive Learning for Lemmatization in Morphology Analysis. In: Fujita H, Selamat A, eds. Intelligent Software Methodologies, Tools and Techniques: SoMeT 2014—Communications in Computer and Information Science, vol 513. Cham, Switzerland: Springer, 2015; 343–357.
10. Lee L. Foundations of Statistical Natural Language Processing. Comput Linguist 2000;26(2):277–279.
11. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Inf Process Manage 1988;24(5):513–523.
12. Sebastiani F. Machine learning in automated text categorization. ACM Comput Surv 2002;34(1):1–47.
13. Rajaraman A, Ullman JD. Data Mining. In: Mining of Massive Datasets. Cambridge, England: Cambridge University Press, 2011; 1–17.
14. Luhn HP. A statistical approach to mechanized encoding and searching of literary information. IBM J Res Develop 1957;1(4):309–317.
15. Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF. J Doc 2004;60(5):503–520.
16. Kalra A, Chakraborty A, Fine B, Reicher J. Machine learning for automation of radiology protocols for quality and efficiency improvement. J Am Coll Radiol 2020;17(9):1149–1158.
17. Chen MC, Ball RL, Yang L, et al. Deep Learning to Classify Radiology Free-Text Reports. Radiology 2018;286(3):845–852.
18. Zech J, Pain M, Titano J, et al. Natural Language-based Machine Learning Models for the Annotation of Clinical Radiology Reports. Radiology 2018;287(2):570–580.
19. Chen PH, Zafar H, Galperin-Aizenberg M, Cook T. Integrating Natural Language Processing and Machine Learning Algorithms to Categorize Oncologic Response in Radiology Reports. J Digit Imaging 2018;31(2):178–184.