

Big Data Computing

Master's Degree in Computer Science

2019-2020

Gabriele Tolomei

Department of Computer Science

Sapienza Università di Roma

tolomei@di.uniroma1.it

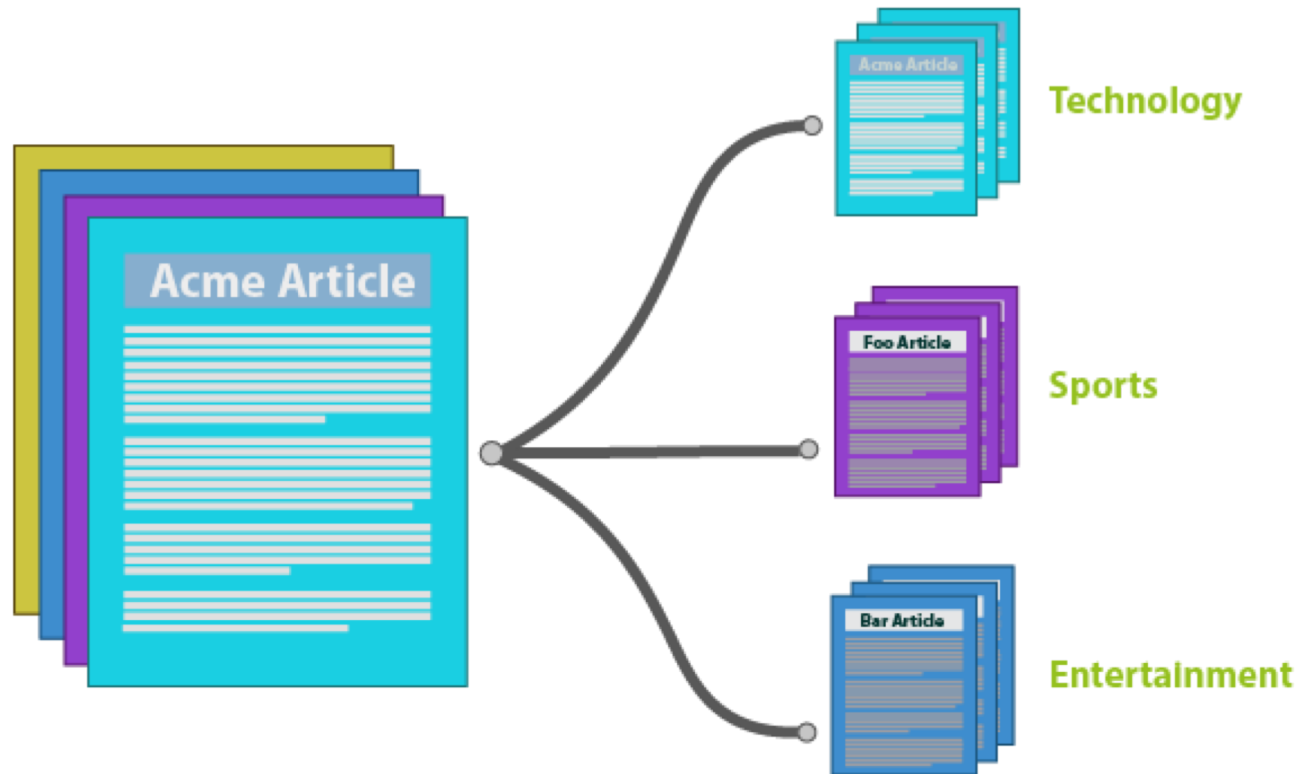


SAPIENZA
UNIVERSITÀ DI ROMA

Recap from Last Lecture(s)

- Dealing with big data requires new computing tools and paradigms
- Hadoop/MapReduce → useful in all those situations where data need to be accessed sequentially
- Spark → general-purpose distributed scalable data processing engine which provides an ecosystem of services to work on (big) data

Our Running Example: Document Clustering



source: <https://towardsdatascience.com/applying-machine-learning-to-classify-an-unsupervised-text-document-e7bb6265f52>

Our Running Example: Document Clustering

- Problem: Group together documents on the same topic
- Documents with similar sets of words may be about the same topic
- Key Issues:
 - Representing documents (in the space of words)
 - Measuring document similarity (in the space of words)

NOTE

A dual problem is topic clustering, where topics (i.e., set of words co-occurring in many documents) are clustered within the space of documents

Document Representation

- Different ways of representing documents (in the space of words)
 - As a set of words (disregarding the order and multiplicity)
 - As a bag-of-words (i.e., a multiset disregarding the order yet keeping multiplicity)
 - As a bag-of- n -grams (i.e., the more general case of bag-of-words)
 - More advanced representations derived from Neural Language Models (e.g., word2vec)
- The choice of document representation affects the similarity measure

Document Representation: Set of Words

doc 1

John likes to
watch movies.
Mary likes
movies too.

{John, likes, to, watch, movies, Mary, too}

doc 2

Mary also likes
to watch
football games.

{Mary, also, likes, to, watch, football, games}

Document Representation: Bag-of-Words

We keep **multiplicity**

doc 1

John likes to
watch movies.
Mary likes
movies too.

```
{  
John:1, likes:2, to:1, watch:1,  
movies:2, Mary:1, too:1  
}
```

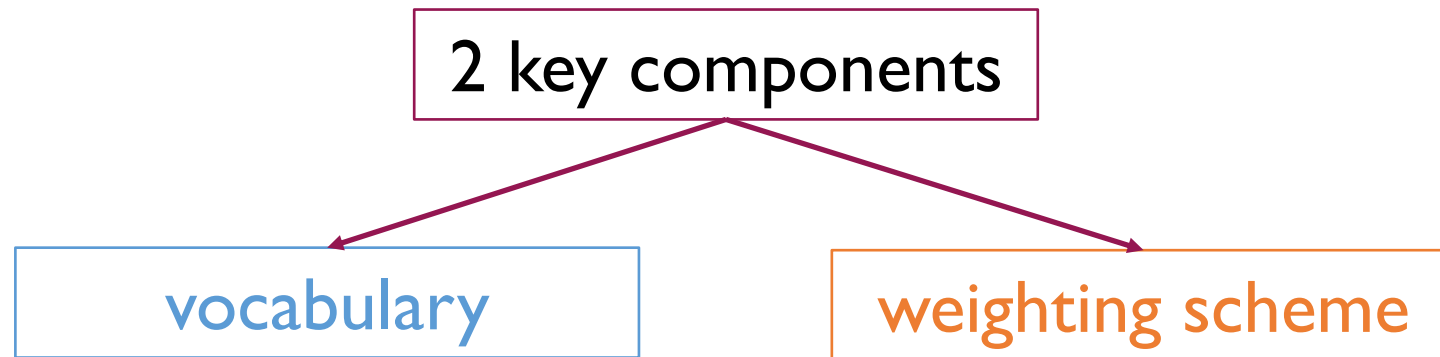
doc 2

Mary also likes
to watch
football games.

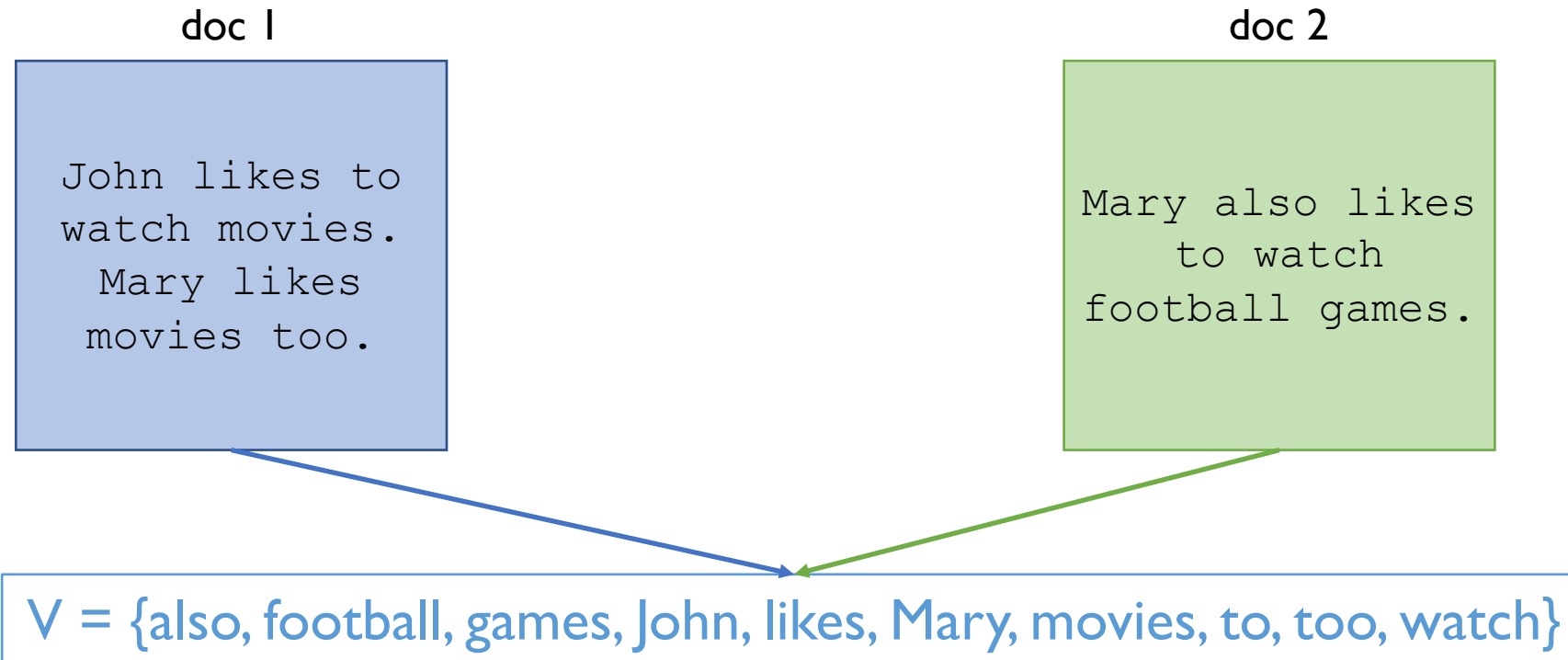
```
{  
Mary:1, also:1, likes:1, to:1,  
watch:1, football:1, games:1  
}
```

Document Representation: Bag-of-Words

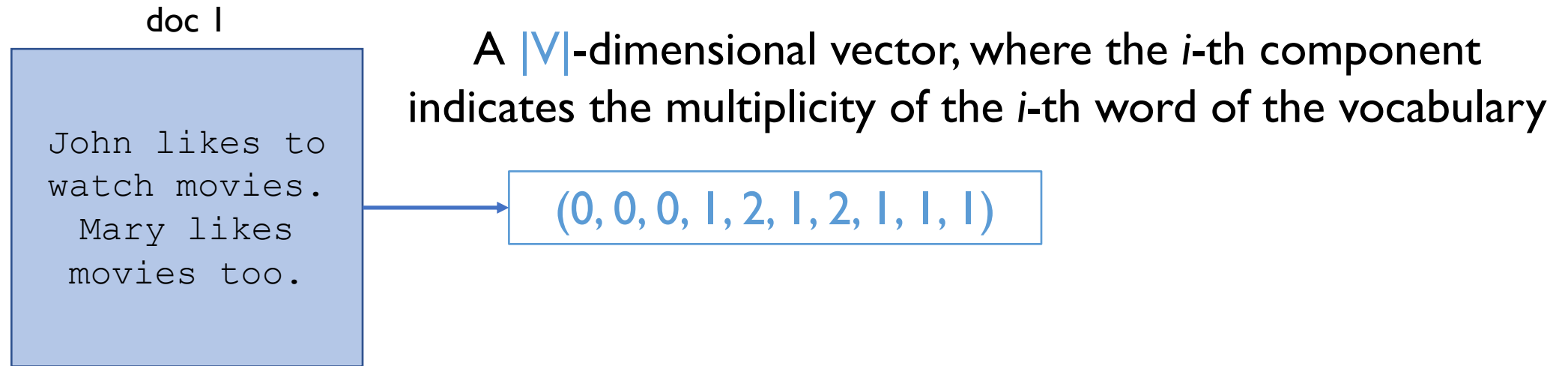
Bag-of-Words (BoW) model is just a preliminary step for more complex document representations



Bag-of-Words: Vocabulary

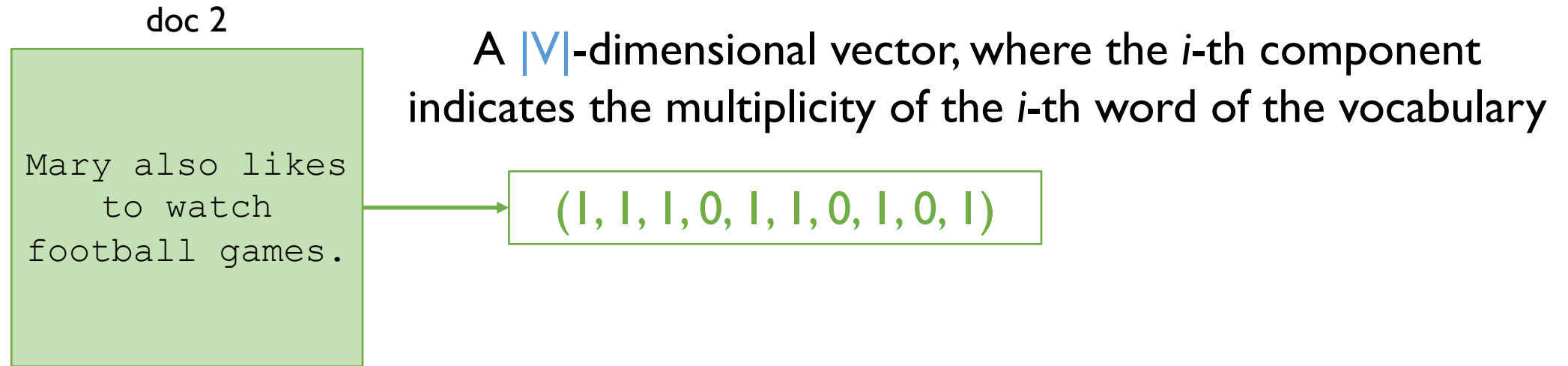


Bag-of-Words: Weighting Scheme



$V = \{\text{also, football, games, John, likes, Mary, movies, to, too, watch}\}$

Bag-of-Words: Weighting Scheme



$V = \{\text{also, football, games, John, likes, Mary, movies, to, too, watch}\}$

Bag-of-Words: A Formal Perspective

$D = \{d_1, \dots, d_N\}$ = collection of N documents

$V = \{w_1, \dots, w_{|V|}\}$ = **vocabulary** of $|V|$ words extracted from D

$\mathbf{d}_i = (f(w_1, i), \dots, f(w_{|V|}, i))$ = $|V|$ -dimensional vector representing d_i

$f : V \times D \mapsto \mathbb{R}$ is a function that maps each word of a document to a real value (**weighting scheme**)

Bag-of-Words: A Formal Perspective

One-Hot (binary) weighting scheme

$$f(w_j, i) = \begin{cases} 1 & \text{if } w_j \text{ appears in } d_i \\ 0 & \text{otherwise} \end{cases}$$

Bag-of-Words: A Formal Perspective

Term-Frequency weighting scheme

$$f(w_j, i) = tf(w_j, i)$$

tf computes the number of times word w_j occurs in document d_i

Bag-of-Words: A Formal Perspective

TF-IDF weighting scheme

$$f(w_j, i) = tf(w_j, i) * idf(w_j)$$

$$idf(w_j) = \log \left(\frac{N}{1 + n_j} \right)$$

Any idea why?

n_j is the number of documents in D containing the word w_j

Bag-of-Words: Limitations and Improvements

- 2 main limitations of BoW model:
 - High dimensionality → sparseness
 - No sequential information nor semantics included → unigram model
- Possible improvements:
 - Use n -grams rather than unigrams to capture sequentiality between consecutive words (i.e., context)
 - Even better, use so-called Neural Language Models like word2vec

Document Representation: Bag-of- n -grams

Example: bigrams ($n=2$)

doc 1

John likes to
watch movies.
Mary likes
movies too.

{"John likes", "likes to", "to watch",
"watch movies", "Mary likes",
"likes movies", "movies too"}

doc 2

Mary also likes
to watch
football games.

{"Mary also", "also likes", "likes to",
"to watch", "watch football", "football games"}

Document Similarity

- We have examined a number of possible document representations
- Depending on those, several similarity measures can be used
- For example, if documents are represented as:
 - set of words \rightarrow Jaccard coefficient
 - one-hot bag-of-words \rightarrow Euclidean distance
 - tf or tf-idf bag-of-words \rightarrow Cosine similarity