Text Processing and Machine Learning



Handling Unstructured Data

Preprocessing of Textual Data Customized WordList Customized Dictionary Tokenizing Stemming Filtering of Tokens Term Frequencies Document Frequencies n-Grams TF-IDF

3 More Text Mining Extensions

Advanced Modeling

Methods for High Dimensional Data Support Vector Machines word2vec Latent Dirichlet allocation (topic modeling) Extract sentiment Text Classification Text Clustering Association Analysis Auto Model Deep Learning Random Forest Gradient Boosting Trees Web Mining

Crawling the Web Transforming Web Sites to Documents Information Extraction

TF-IDF



Term Frequency (TF) is simply the ratio of the occurrence of each word token to the total number of word tokens in the document.

Inverse Document Frequency (IDF) offers another way to look at word tokens relative to the entire corpus than relative to other word tokens in a single document.

np	SENTIMENT ANALYSIS: Detect sentiment in texts using a classification model trained on categor	Process Documents from Da	ata	
	Step 1. Import text data with some assessment of the sentiment related to it. It is processed to extract the words and deliver a word-vector (a numerical representation of the text).	✓ create word vector		٩
		vector creation 💙	TF-IDF •	
	Retrieve Historical S Set Role Nominal to Text Process Documents from Data			
	exa exa ori exa wor	✓ add meta information	Term Frequency Term Occurrences	١
	Main process	🗸 keep text 💙	Binary Term Occurrences	١
	Inside "Process Documents"	prune method 💙	none 🔻	٢
Ρ	rocess Documents from Data	data management	auto 🔻	١
do	C Tokenize Transform Cases Filter Stopwords (English)	select attributes and weights		٩

Latent Semantic Analysis (LSA)



Latent Semantic Analysis is a technique for creating a vector representation of a document. <u>Latent</u> <u>Semantic Analysis</u> takes tf-idf one step further. "Latent Semantic Analysis (LSA)" and "Latent Semantic Indexing (LSI)" are the same thing, with the latter name being used sometimes when referring specifically to indexing a collection of documents for search ("Information Retrieval").



SVD (SVD)

You can inspect LSA results (tf-idf + SVD) for the news feed data by checking the 1st component (SVD_1) in the SVD matrix, and look at the terms which are giving the highest weight (Abstract value of SVD Vector 1) by this component. E.g. Terms from news about U.S. and China trade tariff

Attribute	SVD Vector 1 🤟	SVD Vect
imports	-0.458	-0.113
bn	-0.315	-0.034
china	-0.265	-0.045
wine	-0.249	-0.041

ExampleSet (Process Documents from Data)

word2vec



One of the key problems of text mining is that distances between words are hard to define. How should an algorithm know that "beautiful" and "gorgeous" have the same meaning? Or do they have similar connotations but have different meanings?

Word2Vec is a word vector algorithm which attempts to tackle this problem. An operator takes a word and turns it into a vector. This <u>*Word2Vec* can be associated with the "meaning" of a word</u>. More info: https://community.rapidminer.com/t5/RapidMiner-Studio-Knowledge-Base/Synonym-Detection-with-Word2Vec/ta-p/43860





Latent Dirichlet allocation (LDA)



Topic models provide a simple way to analyze large volumes of unlabeled text. A "topic" consists of a cluster of words that frequently occur together. Using contextual clues, topic models can connect words with similar meanings and distinguish between uses of words with multiple meanings.

LDA operator from operator toolbox extension allows you to identify topics in documents.

Process >	100% 🔎 🔎 📮 🥁 🔝	Extract Topics from Document (L	DA)
ess		number of topics	10
Retrieve Emails data Filter Examples Nominal to Text	Data to Documents Loop Collecti Extract Topics from Document (LDA)	use alpha heuristics	
ori ori	v out top res	alpha	50.0
Read Database	per res	✓ use beta heuristics	
out		top words per topic	30
		iterations	1000
Read Amazon S3			

Entity Recognition



Using the Text mining extension and coupled process documents operators, we can build a process for entity extraction.

Example wordlist created for company names

WordList (Retrieve wordlist_result_entity_company_name)

Word	Attribute Name	Total Occur
19_entertainment	19_entertainment	1
20th_century_fox	20th_century_fox	1
23andme	23andme	1
27b/6	27b/6	1
37signals	37signals	1
3com	3com	1
3m	3m	1 🔫
7-eleven	7-eleven	1
a&m_records	a&m_records	1
a&w_root_beer	a&w_root_beer	1
a_bathing_ape	a_bathing_ape	1
abn_amro	abn_amro	1
ac_motor	ac_motor	1
accenture	accenture	1
acer_inc.	acer_inc.	1
aceralia	aceralia	1
adi_shamir	adi_shamir	1
adidas	adidas	1
adobe_systems	adobe_systems	1

First step is to Process documents from data (e.g. a csv file contains all target names). Using the word list output from step one, we connect it to a Process Documents operator to extract the word list from the text (fracking news).

