Agenda

• Text Mining Goals and Usage
• Enrichment & Preprocessing
• Data Types & Structures
• Visualization
• Topic Detection
• Sentiment Analysis
Install TextProcessing Extension

KNIME:

www.knime.org

Install Textprocessing Extension under KNIME Labs
Examples

Example Workflows available on the KNIME public server.
Text Mining Workflow

Create New Document → Enrichment → Transformation → Preprocessing

Filtering, Stemming, ...

Bow

Frequencies

TF abs, TF rel, , IDF, ...

Visualization

Tag Cloud and Document Properties Visualization

Data Mining

Classification for Topic Detection
1 - Create a Document
New Data Types

Document
Encapsulates text, author, title, source, category, and type
From a Folder

The output is a list of Documents
From PUBMED

The output is a list of Documents

Destination Folder MUST be EMPTY!
Strings to Document

Diagram showing the workflow of reading data from TripAdvisor, filtering rows with missing restaurant names, and creating documents from string texts.
From RSS Feeds

Download of latest NY Times RSS feed and transforming into document cells

Table Creator ➔ HttpRetriever ➔ FeedParser ➔ Strings To Document

NY Times rss feed ➔ download data from an http address ➔ Parses and RSS feed ➔ converting RSS feeds to Documents

http://feeds.nytimes.com/nyt/rss/World
The Data Set

Reviews of Restaurants in Berlin from TripAdvisor

Self-downloaded with RSS Feeder
2 - Enrichment
New Data Types

Document
Encapsulates text, author, title, source, category, and type

Term
Encapsulates a term
Enrichment (Tagging)

Enrichment nodes (mostly) change the granularity of terms.

- Multiword detection, named entity recognition, part of speech definition, ...
- To each detected entity (term) a tag is added, specifying its type.
- To avoid intersection of granularity the last node dominates.
Tagger Conflict Resolution

In case of intersections of granularity the last node overwrites.
Example: “The gene interleukin 6 interacts …. ”
1. POS tagger: “The\DT gene\NN interleukin\NN 6\CD interacts \VBZ ”
2. NE tagger: “The\DT gene\NN interleukin 6\GENE interacts \VBZ ”

POS Tagger

NE Tagger

Overwrite!

Adds POS tags to terms.

Adds NE tags to terms and overrides other conflicting tags.
Tagging

ENRICHMENT
Part Of Speech (POS) Taggers
POS tagger  Stanford tagger

Penn Treebank Tags  Stanford Tags for engl, deu, frch

Named Entities Taggers
OpenNLP NE tagger  Abner tagger  Oscar tagger

Named Entities Locations, Organizations, Persons, Money, Date, Time
biomedical named entities, as: genes, proteins, ...
chemical named entities

Custom Taggers
Dictionary tagger  Wildcard tagger

custom tags from a specified dictionary column
tags words according to a wildcard or regular expression pattern

Named Entity Tags can be set as unmodifiable

Dictionary Column
Tag Type
Tag Value

Language + POS Model
Abner Model
No settings

No settings

Dictionary Column
Tag Type
Tag Value

Matching Strategy

KNIME Copyright 2013
KNIME Text Mining Webinar
Unmodifiable Named Entity Tags

Named Entities Tags attached through enrichment nodes can be set as unmodifiable

Unmodifiable Tags are not affected by any preprocessing nodes (stemming, filtering, etc.)
Workflow

Strings to Document Workflow + POS Tagger
3 - Transformation
Data Types and Features

- String
- Document
- Tags
- Meta Info

- Sentence
- Term
- Molecule Structure
- Document Vector
Parsing / Tokenization

Parser nodes parse documents by applying standard tokenization via OpenNLP tokenizer.

Each token is a term consisting of a single word.

Tags are applied to terms.
Annotations label a section part in the document (like “Abstract”, “Title”, etc.)

Terms consist of tokens and tags.
The Bag of Words

- No Settings required
- Each Term is extracted with Tags (NN, VB, …)
- List (Bag) of Terms (Words) identified in Document
Data and Sentence Extractor
Conversions

Molecular Structure
Workflow

**Table Reader**
- read data
- food from TripAdvisor

**Row Filter**
- no rows with missing restaurant name

**Strings To Document**
- create documents

**Column Filter**
- keep docs only

**ENRICHMENT**
- POS tagger
  - attach Tags for Part Of Speech

**TRANSFORMATION**
- BoW creator
  - creates Bag Of Words from document
4 - Preprocessing
Preprocessing Mode

Normal
- Faster
- Preprocesses only terms of the term column
- Documents are not changed

Deep
- Slower
- Preprocesses terms of the term column
- Terms in documents are changed as well
- Unchanged documents can be appended
Filtering by Tags and Terms

- biomedical named entity tags
- French Treebank (POS) tags
- Any tags
- numbers, words <N chars, modifiable terms
- POS, chemical, and pharma tags
- Remove Punctuation from document
- RegEx terms
- STTS tags
- Standard Named Entity Filter tags
- Stop word terms
Converting and Replacing

- Term case conversion
- Term replacement with dictionary word
- Introducing Hyphenation (Liang’s algorithm)
- RegEx based term replacer
- Groups rows by term value
Stemming

- Kuhlen Stemmer (English only)
- Porter (English only)
- Snowball (English, German, French, ...)

Stemmed term replaces original term!

convert → converting → convert[]
Workflow

Table Reader → Row Filter → Strings To Document → Column Filter
- Table Reader: read data from TripAdvisor
- Row Filter: no rows with missing restaurant name
- Strings To Document: create documents
- Column Filter: keep docs only

ENRICHMENT
- POS tagger: attach Tags for Part Of Speech

TRANSFORMATION
- BoW creator: creates Bag Of Words from document

PREPROCESSING
- Number Filter: filter Numbers
- Punctuation Erasure: remove punctuation
- Stop word Filter: remove stop words (and, but, is ...)
- Case converter: all lowercase
- Snowball Stemmer: English Stemmer from the Snowball Library
- RegEx Filter: remove strange chars sentences
- POS Filter: keep adjectives (JJ*), nouns (NN*), and verbs (VB*)
5 - Frequencies
Frequency Measures

- **TF**
  - Term Frequency
  - TF absolute = # occurr. of term \( t \)
  - TF relative = # occurr. of term \( t \)/ # terms

- **IDF**
  - \( \text{IDF} = \log(1 + \frac{\# \text{ docs}}{\# \text{ docs with term } t}) \)

- **ICF**
  - \( \text{ICF} = \log(1 + \frac{\# \text{ cat.}}{\# \text{ cat. with term } t}) \)

- **IDF * TF**
Frequency Filter

Values between min threshold and max threshold

Frequency Column

Top K Rows

Min max thresholds

K

Number Settings

Number 1.000

Filter column D TF rel

Filtering by

Threshold

Number of terms

Min Max Settings

min= 0.01

max= 1

Workflow
6 - Visualization
Document Viewer

Right-click word list search engines

Search engines listed in KNIME->Textprocessing->Search Engine Preferences

Double-click opens the document

Double-click opens the document.
Document Viewer

Previous and next document

Amazing burgers! Amazing!!!

If you like burgers you have to visit here! Very much USA style. The food is stunning, great staff and all great value. I would highly recommend booking as it was packed at 10pm on a Monday evening. Seriously yummy!!!
Tag Cloud

Adj, verbs and nouns as same word
Asian Restaurants
German Food Restaurants

beer

Berlin Food Restaurant

deli
trad

look
cook

decor
cuisine

waiter
tasting

menu

service

nice

meal

eat

reservation

serve

dish

perfect
disappointment

attractor

eak

dark

inform

follow

window

e-mail

pay

accept

the

speak

german

style

be

deer

situat

eat

bicycle

distance

free

satisfaction

history

goulash

describe

cooking

own

wonder

reserve

help

portion

step

warm

impress

told

awson

session

vari

parti

accent

through

fulfill

sufficient

stuff

bowls

per

differ

skeptical

fire

brew

car

north

strange

pool

egg

heart

weak

breast

bone

salad

with

bread

play

stop

quiet

allow

sun

paid

mustard

sort

research

due

space

wide

facil

samos

friendlier

speed

faster

contemporary

flight

prove

regret

afternoon

avoid

start

one

birthday

complain

treat

neighborhood

good

enter

mastercard

study

rest

cray

restaurant

succeed

creation

snow

glass

start

half

dancer

friends

steak

coves

chance

sold

serve

search

ham

type

tongue

student

realize

photo

paper

sequence

account

wrap

get

off

food

happier

tag

lock

extens

创造性

level

survive

flavor

flawless

gendar

spoil

realize

pianist

commit

proper

screw

ad

ceman

goulash

skeptical

internet

hot

test

option

pass

uk

fit

spit

quint

circus

benefit

reign

flavor

lemonade

transit

kid

kiss

off

man

leaf

blind

patient

mount

southern

forest

get

light

knife

hand

land

appear

overhead

near

pancake

furniture

bed

violin

klap

occur

among

meat

in

tares

jeweler

lighthouse

white

mit

juicelibrary

KNIME Copyright 2013

KNIME Text Mining Webinar

46
Fast Food Restaurants
Hiliting in Tag Clouds

Interactive Table
7 - Topic Classification
Document Vector

Document Vector: Documents represented in the terms space

Bitvector or frequency measure
Term Vector: Terms represented in the documents space

Bitvector or frequency measure
Topic Detection Goal

Possible Topics:
- Asian Restaurants
- German Food
- Fast Food

Pre-labeled data set available!
Target Topic is in Document Category.
After the Document Vector Transformation, topic detection becomes just another data analytics problem.

80% for training set, 20% for test set.
Target = Category
Classification Sub-Workflow

Prepare input data as word in Document 1/0

Extract Category

Training/Testing

Wrong docs

X-Validation
Problem 1

Sometimes a pre-labeled data set is not available.

1. Use a Clustering technique
2. Find a similar pre-labeled data sets that you can adapt to the current problem
Problem 2

The vector generated by the Document Vector node can be high dimensional. To reduce the input space dimensionality you can:

- Filter words by frequency
- Detect keywords and only use the most important ones.
Keywords Extractor Nodes

From: "KeyGraph: Automatic Indexing by Co-occurrence Graph based on Building Construction Metaphor" by Yukio Ohsawa.

From: "Keyword extraction from a single document using word co-occurrence statistical information" by Y.Matsuo and M. Ishizuka.
8 - Sentiment Analysis
Sentiment Corpus

- MPQA Corpus with negative and positive words
- Tag Words according to Corpus with a Dictionary Tagger Node
Dictionary Tagger Node

List of Words from Corpus

Tag Type to attach to Words in document and in Corpus List

NE = Named Entities

Tag Value to attach to Words in document and in Corpus List

Each Tag Type has a list of possible Tag Values
**Sentiment By Document**

**PERSON** Tag for POSITIVE Words  
**TIME** Tag for NEGATIVE Words

**ATTITUDE** = +/-1 for POSITIVE or NEGATIVE Words

**MEAN(ATTITUDE)** on all words for each document

Bin1 = negative documents  
Bin2, Bin3 = neutral documents  
Bin4 = positive documents

Adjust Bin boundaries in Auto-Binner node for more accurate results.
Sentiment by Category

- Tag Cloud on all words in all documents for a given category
- Words in tag clouds are colored by sentiment

New column NE with values: PERSON, TIME, neutral (when missing value)
Sentiment by Word

• Find Most Positive/Most Negative Document
• Build Tag Cloud with words colored by sentiment
Tag Cloud of Worst/Best Doc

Most Positive

- beautifully
- food
- attention
- friendly
- price
- eat
- reason
- visit
- prepare
- staff
- wonderful
- twice
- help
- stay
- meal
- lovely
- delicious

Most Negative

- drink
- expensive
- poor
- burger
- price
- tasty
- New York
- service
- unconsidered
- staff
- fine
- slack
- London
- head
- better
- cheap
Improvements

- Add polarity change for negations
- Add polarity changes for enhancements
- Improve positive/negative dictionary
- Improve bin distribution (skeweness towards positive)
- Improve list of stop words
- Remove Names of Burgers?
Thank you

education@knime.com