Text Mining and Processing: From Foundations

Jacob Coles @ Redfield AB
Modified from KNIME AG content

You are free to:
Share
copy and redistribute the material in any medium or format
Adapt
remix, transform, and build upon the material for any purpose, even commercially.

You must give **appropriate credit**, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

WELCOME!
VÄLKOMMEN!
WELKOM!
BIENVENUE!

REDFIELD

KNIME Ready

Trusted Partner

OVERVIEW

Today's focus: Fraud/Anomaly detection

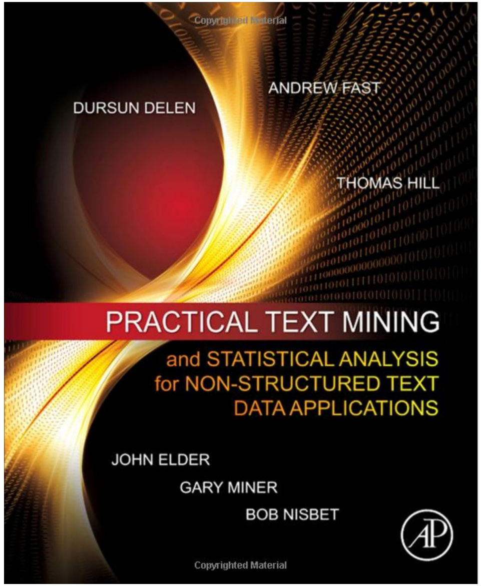
Wifi: guest_hr@hr
Password: guest_hr

Link to Knime Hub Workflows:
https://t.ly/7I2mo
or hub.knime.com/jacobcoles

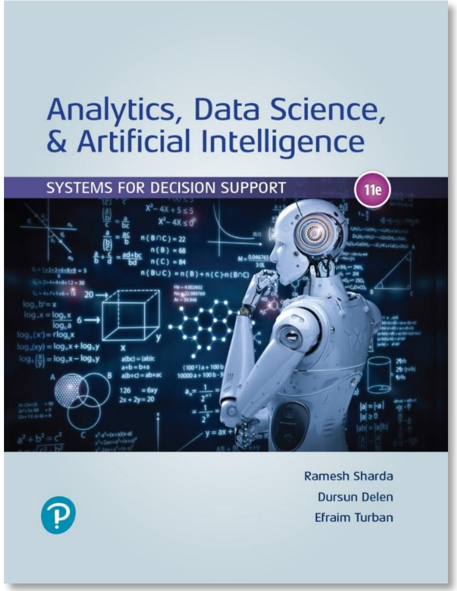
Session 1
Introduction,
Importing Text,
Elementary Processing

Sources / References

Chapter 7 – Text Mining, Sentiment Analysis, and Social Analytics



© 2012



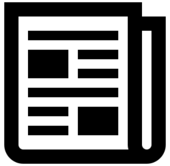
© 2020

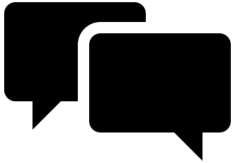
+
Articles
White papers
Tutorials

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

There Are Many Terms

- Text Mining
- Text Analytics
- Text Processing
- Information Retrieval
- Information Extraction
- Natural Language Processing
- Computational Linguistics
- Unstructured Data Mining
- …





These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Why Text Mining?

- Roughly 85-90 percent of all corporate data is in some kind of unstructured form (e.g., text)
  - What does unstructured really mean?
- Unstructured corporate data is doubling in size every 18 months…
- Tapping into these information sources is not an option, but a necessity to stay competitive
- Text IS data

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

What Is Text Mining?

- Definition of Text Mining:
  - Extraction of useful information from unstructured text sources.
  - Examples include emails, social media posts, documents, and journals.
- Contrast with Traditional Data Mining:
  - Traditional data mining focuses on structured data from databases and spreadsheets.
  - Text mining deals primarily with unstructured text.
- Challenges of Text Mining:
  - Text is inherently unstructured and variable.
  - Difficulty in generalizing text for computational analysis without conversion to numeric formats.
- Key Process:
  - Involves transforming text into a format amenable to computational tools and analysis.

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

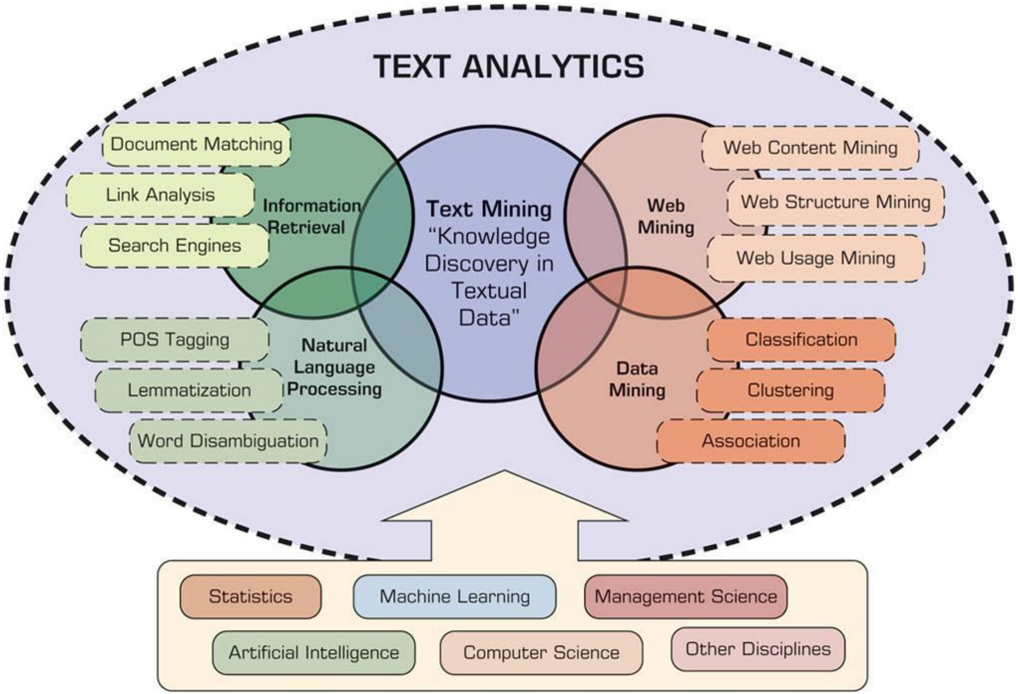
Example Use-Cases

- Anomaly Detection
  - Spotting unusual patterns
  - Security breaches, fraudulent transactions
- Law
  - Automating document reviews
  - Case predictions
- Academia
  - Analyzing research papers
  - Predicting trends, identifying key themes
- Marketing
  - Understanding consumer sentiment
  - Informing decision making from social media, feedback forms
- Spam Filtering and Prioritization
  - Managing and prioritizing vast quantities of communications

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Text Mining versus Text Analytics

- There are many sub-topics in this field
- Text-mining, analytics and data-mining are all related
- We want to gather insights from ALL our data



Copyright © 2020 by Pearson Education, Inc.

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Introduction to Text Mining in Fraud Detection

- Objective in Fraud Detection
  - Uncover hidden patterns and anomalies
  - Detect irregularities in large data sets
- Process Overview
  - Transform raw text into structured format
  - Perform sophisticated analyses
- Key Concept
  - Initial step: Break text into smaller pieces (tokens)
  - Analyze frequency and context of words to spot patterns
  - Example: Frequent mentions of "refund" or "delay"

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Text Mining Application – Fraud Detection

Number	Construct (Category)	Example Cues
1	Quantity	Verb count, noun phrase count, etc.
2	Complexity	Average number of clauses, average sentence length, etc.
3	Uncertainty	Modifiers, modal verbs, etc.
4	Nonimmediacy	Passive voice, objectification, etc.
5	Expressivity	Emotiveness
6	Diversity	Lexical diversity, redundancy, etc.
7	Informality	Typographical error ratio
8	Specificity	Spatiotemporal information, perceptual information, etc.
9	Affect	Positive affect, negative affect, etc.

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Basic Process of Text Mining

- Starting with the Basics
  - Elementary techniques for foundational understanding
  - Importance of conceptualizing the process
- Initial Steps in Text Mining
  - Step 1: Importing and viewing text
  - Step 2: Preprocessing; Cleaning and tokenization
  - Step 3: Vectorization; Creating term-document matrix
  - Step 4: Extracting knowledge; Analysis and insights

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Step 1 - Importing Text

- Collecting the Corpus
  - Gathering documents and data
- Importing Methods in KNIME
  - Flat File Document Parser
    - Extract text from all document types (basic structure)
  - Microsoft Word/Excel Parsers
    - Extract text from Word and Excel files
  - PDF Parser
    - Extract text from PDF documents
  - Document Grabber
    - Fetch and extract text from various sources
  - TIKA Parser
    - Identify and extract text from a variety of file types

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Step 2 - Preprocessing

- Preprocessing Steps
  - Convert unstructured text to analyzable form
  - Tokenization

    Break text into smaller pieces (tokens)

    Example: "Please approve the attached invoice" → ['Please', 'approve', 'the', 'attached', 'invoice']
- Importance of Tokenization
  - Computers need text to be in quantifiable, analyzable pieces
  - Each token becomes a building block for further analysis

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

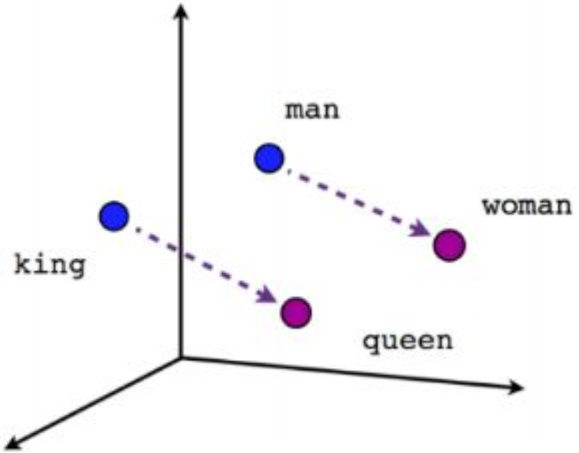
Step 3 - Vectorization

- Storing Tokens in a Useful Way
  - Creating a Term-Document Matrix
    - Transform text into numerical format
    - Represent documents as numerical data
- Example Process
  - Count instances of words in documents
  - Visualize vectors in a tabular format

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Extracting Knowledge from Vectors

- Understanding Vectors
  - Measure similarity between documents
  - Use metrics like cosine similarity or Euclidean distance
- Thought Experiment
  - Visualizing document vectors in space
  - Dimensionality reduction for analysis



These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Step 4: Analysis Techniques

- Methods of Analysis
  - Distance Metrics: Euclidean distance, cosine similarity
  - Clustering and Classification
  - Anomaly Detection for fraud detection
- Visualizing Data
  - Bar charts for term frequency
  - 2D/3D graphs for clusters using dimension reduction

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Example and Demonstration

- Email Analysis
  - Create term-document matrix
  - Simple analysis to detect potential fraud
- Visualisation Techniques
  - Bar charts for term frequency
  - Clusters in space using dimension reduction
- KNIME Demo
  - Importing emails using PST Reader
  - Visualizing and classifying data

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Other Preprocessing Steps

- Stop Word Filter
  - Removes common, low-value words
  - Examples: "and", "the", "is"
- Tag Filter
  - Selectively processes words based on parts of speech
  - Focused analysis on nouns, verbs, adjectives
- Stemmer
  - Cuts off any ‘grammatical endings’
  - Simplifies analysis by consolidating similar words
- Lemmatization
  - Reduces words to their root form
  - Also simplifies analysis by consolidating similar words

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Use of Text Mining

Stemming vs. Lemmatization

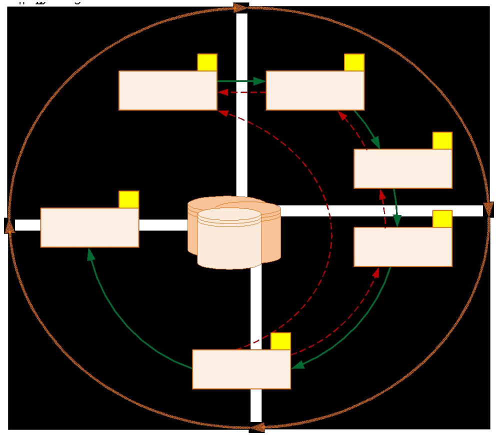
- Common goal: to generate the root form of the words
- Reason: to merge variations of the same words together
- Difference:
  - Stem results in truncated/chopped words (not necessarily a complete word)
    - Stemming is syntactic and fast - follows an algorithm where ends of words are cut off
      - Original word: Running -> Stemmed form: Run
      - Original word: Better -> Stemmed form: Bett
  - Lemma results in an actual language word (inflection free)
    - Lemmatization is semantic and slower as it follows a linguistic dictionary
      - Original word: Running -> Lemmatized form: Run
      - Original word: Better -> Lemmatized form: Good

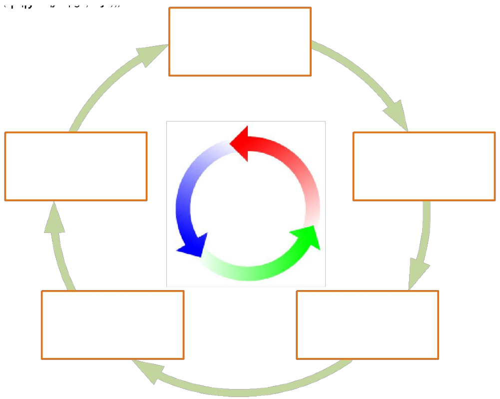
These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

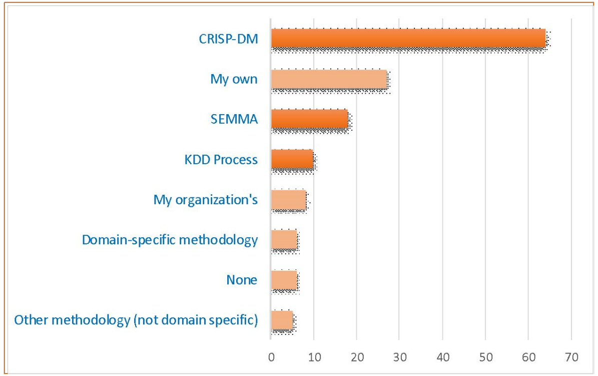
Text Mining Process
A Higher Level Approach

Text Mining Process

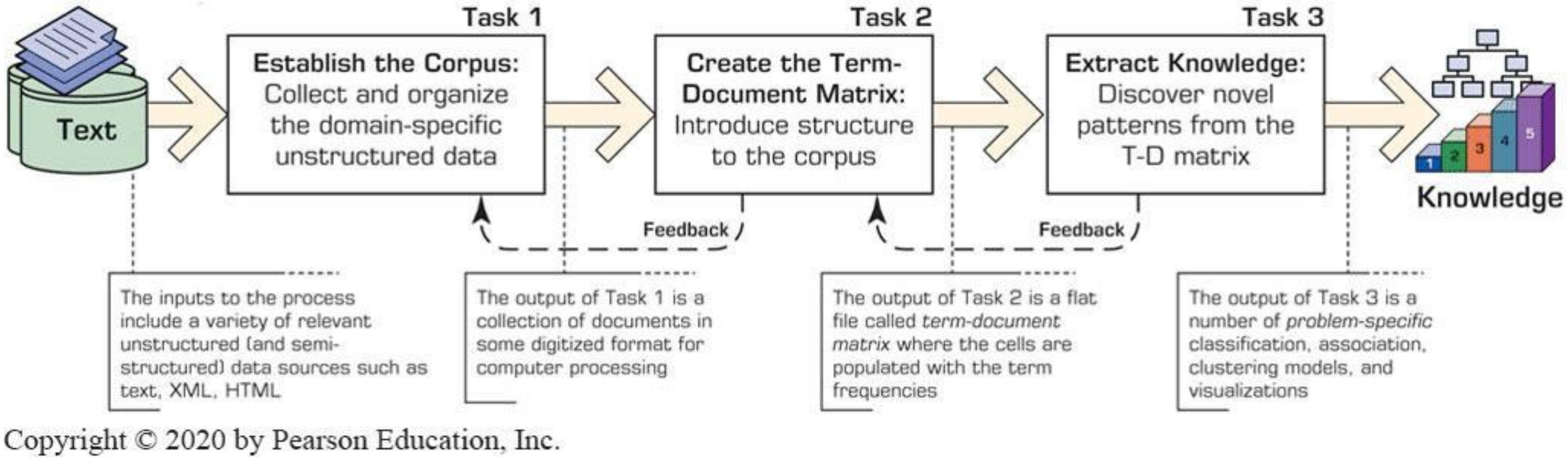
- A standard process: the manifestation of the “best” practices
- Standard process for data mining:
  - Cross industry process for data mining (CRISP-DM)
  - Sample, Explore, Modify, Model, Assess (SEMMA)







Text Mining Process



These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Text Mining Process

- Task 1: Establish the corpus
  - Collect all relevant unstructured data (e.g., textual documents, XML files, emails, Web pages, short notes, voice recordings…)
  - Digitize, standardize the collection (e.g., all in ASCII text files)
  - Place the collection in a common place (e.g., in a flat file, or in a directory as separate files)

- Task 2: Create the Term–by–Document Matrix (TDM)
  - Should all the terms be included?
    - Stop words, include words
    - Synonyms, homonyms
    - Stemming, lemmatization
  - What is the best representation of the indices (values in cells)?
    - Row counts; binary frequencies; log frequencies
    - TF/IDF

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Text Mining Process

- Task 3 Create TDM
  - TDM is a sparse matrix. How can we reduce the dimensionality of the TDM?
  - Manual by a domain expert, frequency based, SVD, …

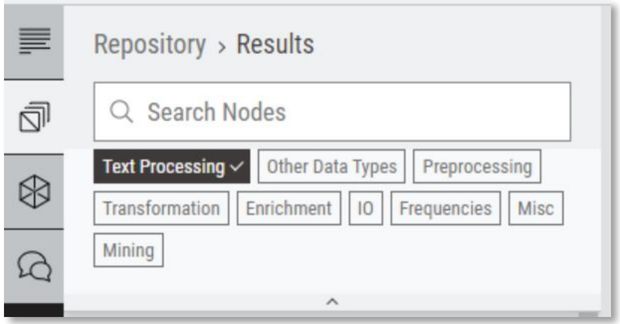
Terms Documents	Investment Risk	Project Management	Software Engineering	Development	SAP	...
Document 1	1			1		
Document 2		1				
Document 3			3		1	
Document 4		1				
Document 5			2	1		
Document 6	1			1		
...						

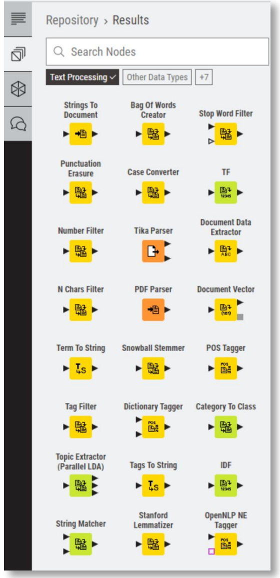
- Task 4: Extract knowledge
  - Classification (text categorization)
  - Clustering (natural groupings of text)
    - Improve search recall
    - Improve search precision
    - Scatter/gather
    - Query-specific clustering
  - Association
  - Trend Analysis

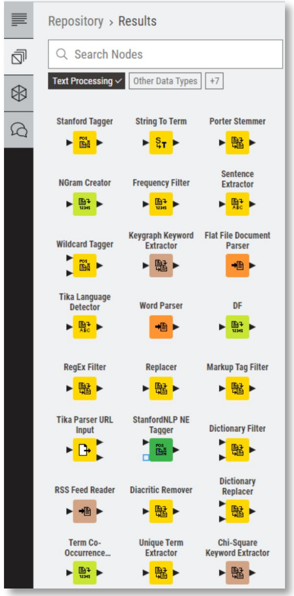
These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Text Mining Process in KNIME

- Logical organization of KNIME Text Processing nodes
  - Look for the tag “Text Processing” in the Node Repository
  - Filter nodes by tags, e.g., “Enrichment”, “Frequencies”, etc.







These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Advanced Preprocessing with Redfield Spacy Nodes

- Similar to Knime Textprocessing but updated tools based on machine learning
- Nodes
  - Tokenizer
    Splits text into individual words or tokens
  - NER (Named Entity Recognition)
    Identifies and classifies key entities
  - POS Tagger (Part of Speech)
    Assigns grammatical roles to words
  - Lemmatizer
    Refines words to their dictionary form
  - Morphologizer
    Analyzes word formation and structure
  - Stop Word Filter and Vectorizer
    Further text cleaning and numerical conversion

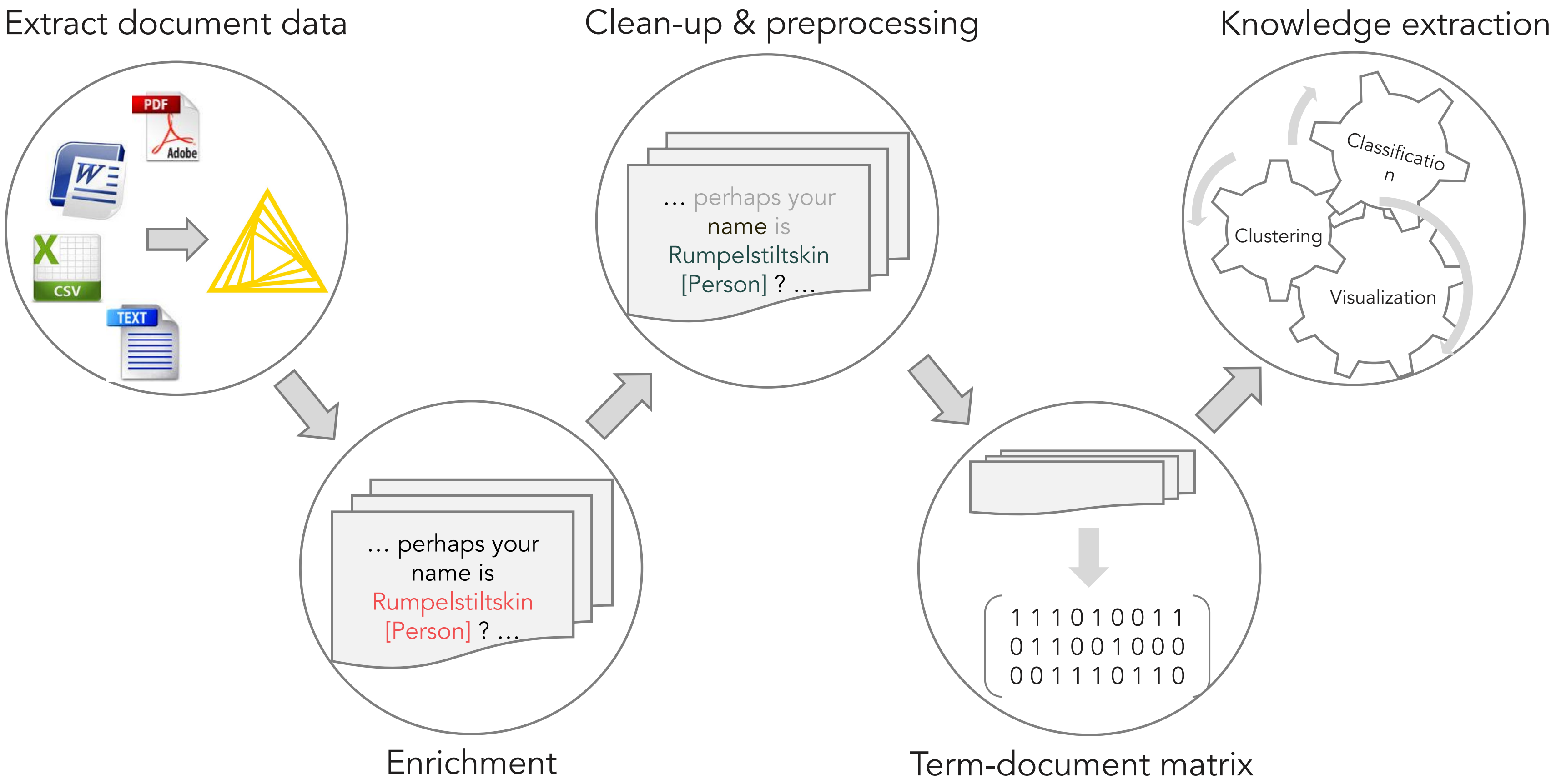
These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

If we have time:
Explore Some Knime Workflows

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Session 2
Advanced Data Mining

Recap



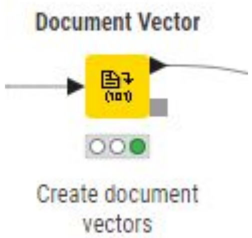
These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Key Node: Document Vector (Nodes)

- Transforms bag of words into document vectors
  - Creates numerical vectors from ‘Terms’
  - We first create terms from the previously shown methods

Bag of words with frequency column

Term Term	Document Text document	Preprocess... Text document	TF rel Number (double)
build[NN(POS)]	"Who Doesn't like I...	"italian"	0.053
columbu[NNP(PO...	"Who Doesn't like I...	"italian"	0.053
histori[NN(POS)]	"Who Doesn't like I...	"italian"	0.053
italian[NNP(POS)]	"Great Italian Foo...	"italian food serv i...	0.133
food[NNP(POS)]	"Great Italian Foo...	"italian food serv i...	0.067
serv[VBN(POS)]	"Great Italian Foo...	"italian food serv i...	0.067
peopl[NNS(POS)]	"Great Italian Foo...	"italian food serv i...	0.133



Document vector

Document Text document	restaur Number (do...	ladi Number (do...	suggest Number (do...
"idea restaur ladi hop suggest restaur ti...	1	1	1
"advic chanc fridai wait busier meal staff...	0	0	0
"italian restaur citi spinach pizza chicken...	1	0	0
"nice restaur live food price clean peopl f...	1	0	0
"love meal night servic superb food ama...	0	0	0
"amaz food staff wonder time amaz"	0	0	0
"third time restaur time locat washington...	1	0	0

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

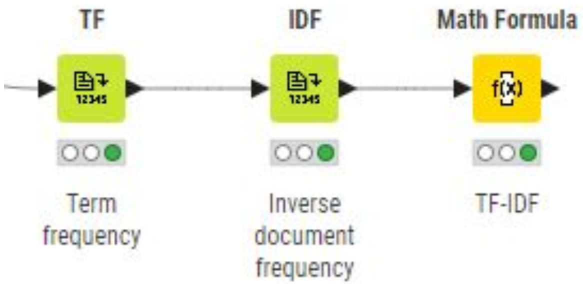
Enhancing Term-Document Matrix with TF-IDF

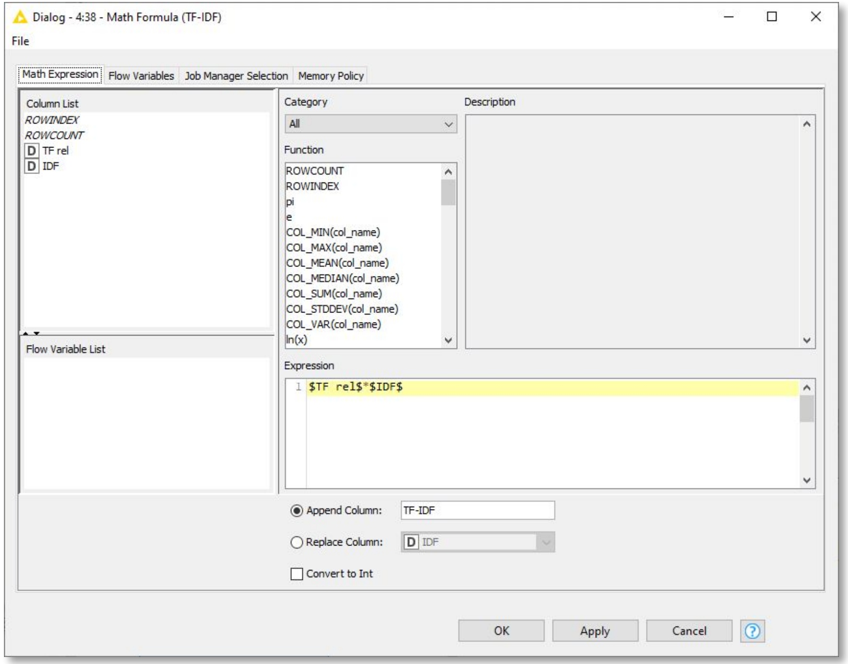
- Term Frequency (TF)
  - Measures term frequency in a document
  - Insight into term importance within the document
- Inverse Document Frequency (IDF)
  - Evaluates term rarity across documents
  - Distinguishes unique terms
- TF-IDF
  - Combines TF and IDF
  - Highlights distinctive words in each document
- Application Example
  - Using KNIME to calculate and visualize TF-IDF

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Combination of Nodes: TF-IDF

- Multiplies relative TF with IDF to measure importance of term





Data Exploration and Dimension Reduction

- Vectors and Patterns
  - Extract features like word frequencies and tags
  - Use term frequency matrix for document classification and outlier detection
- Dimension Reduction
  - Simplifies high-dimensional data
  - Makes analysis and visualization easier
  - Preserves key properties of vector closeness
- Thought Experiment: World Globe to Map
  - Reduces dimensions while preserving essential relationships

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Machine Learning Models for Text Analysis

- Vector Representations
  - Transform text into numerical format
  - Use vectors for analysis and ML model training
- Types of Models
  - Decision Trees: Sequential decision making
  - Neural Networks: Pattern recognition through layers
  - Naive Bayes: Probability-based classification
  - Logistic Regression: Classification method
  - Support Vector Machine: Data point mapping and separation
  - Tree Ensembles: Enhanced reliability and accuracy

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Use Cases

- Outlier detection
- Sentiment analysis
- Clustering
- Outlier detection

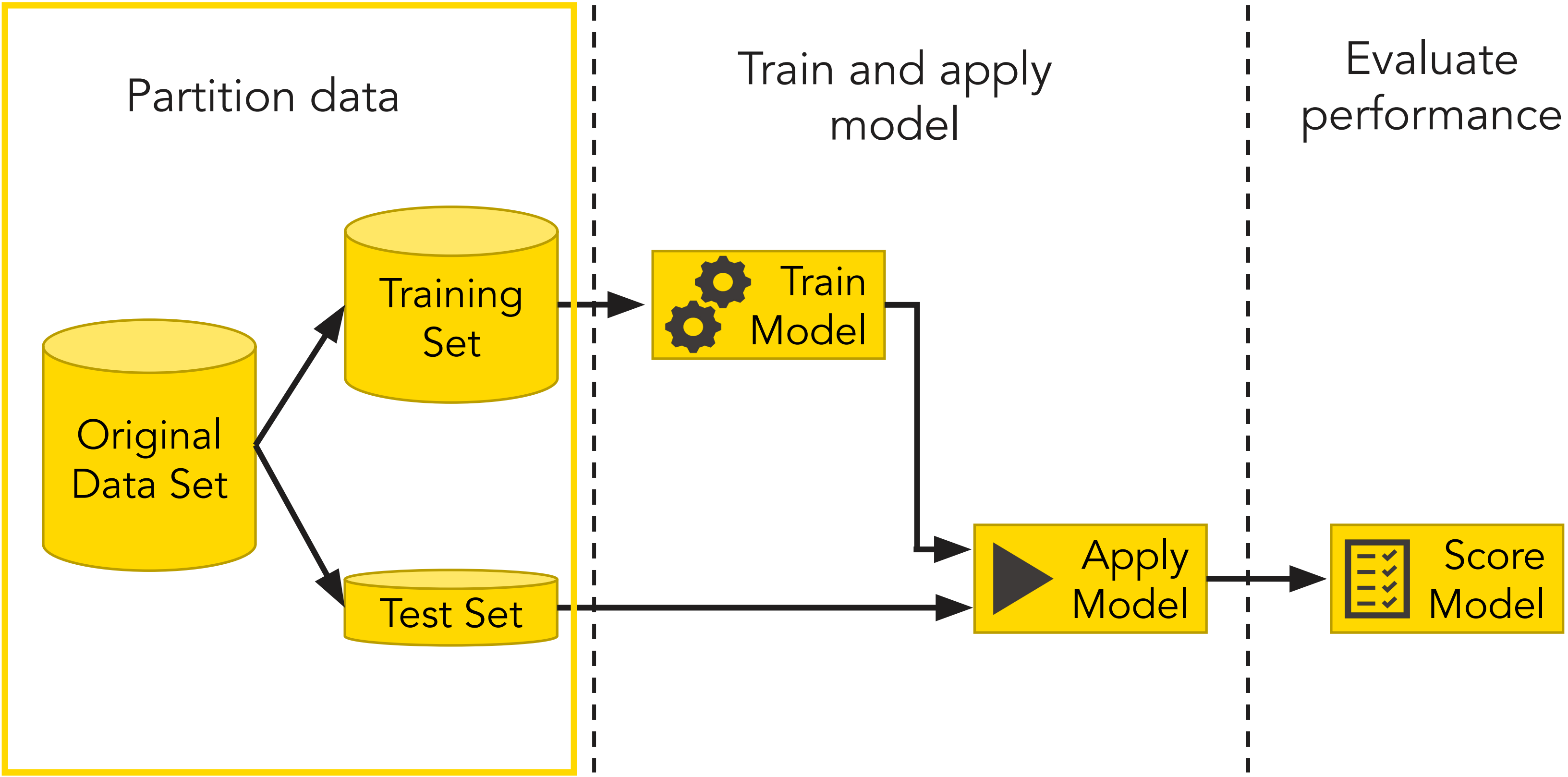
These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Train-Test Split and Model Evaluation

- Train-Test Split
  - Divides data into training and testing sets
  - Ensures realistic model performance testing
- Partitioning Node in KNIME
  - Efficient data partitioning
- Confusion Matrix
  - Visualizes model performance
  - Identifies misclassification patterns
- Accuracy Measures
  - Precision, Recall, F1-Score
  - Detailed view of model reliability

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

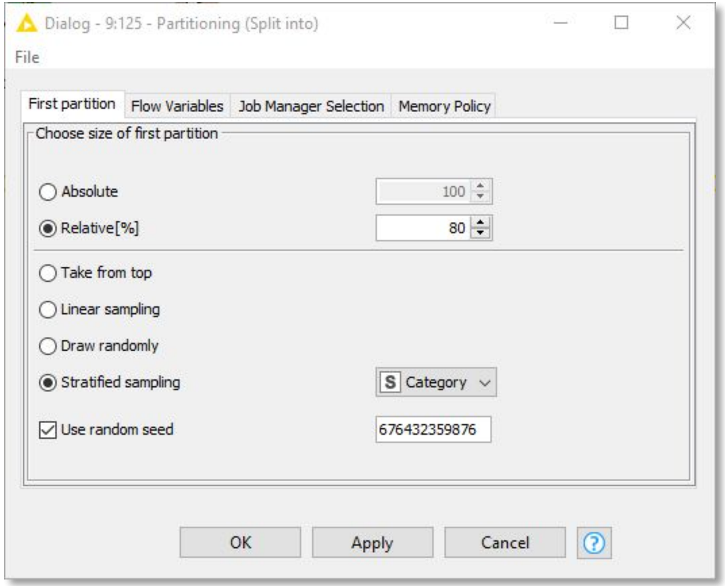
Data Mining: Process Overview



These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Node: Partitioning

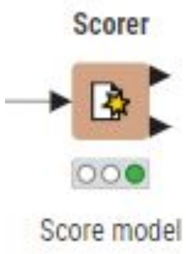
- Use it to split data into training and evaluation sets
- Partition by count (e.g. 10 rows) or fraction (e.g. 10%)
- Sample by a variety of methods; random, linear, stratified

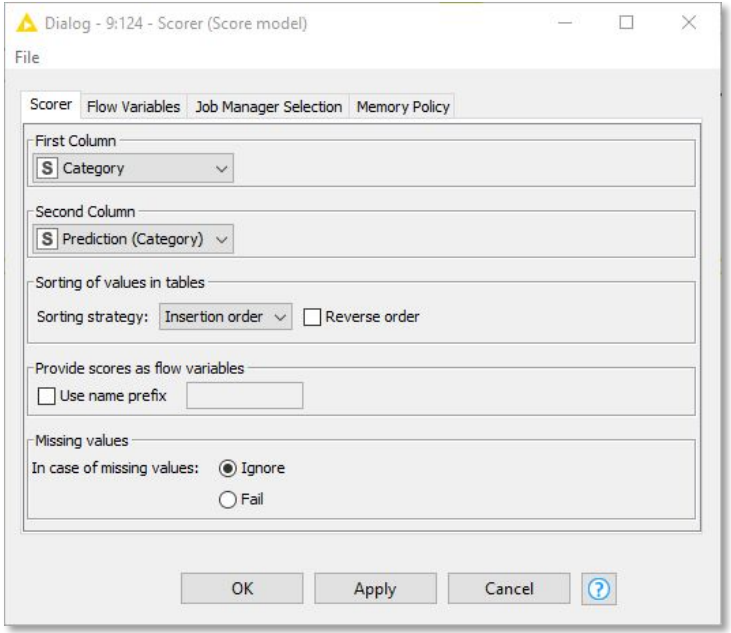


These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Node: Scorer

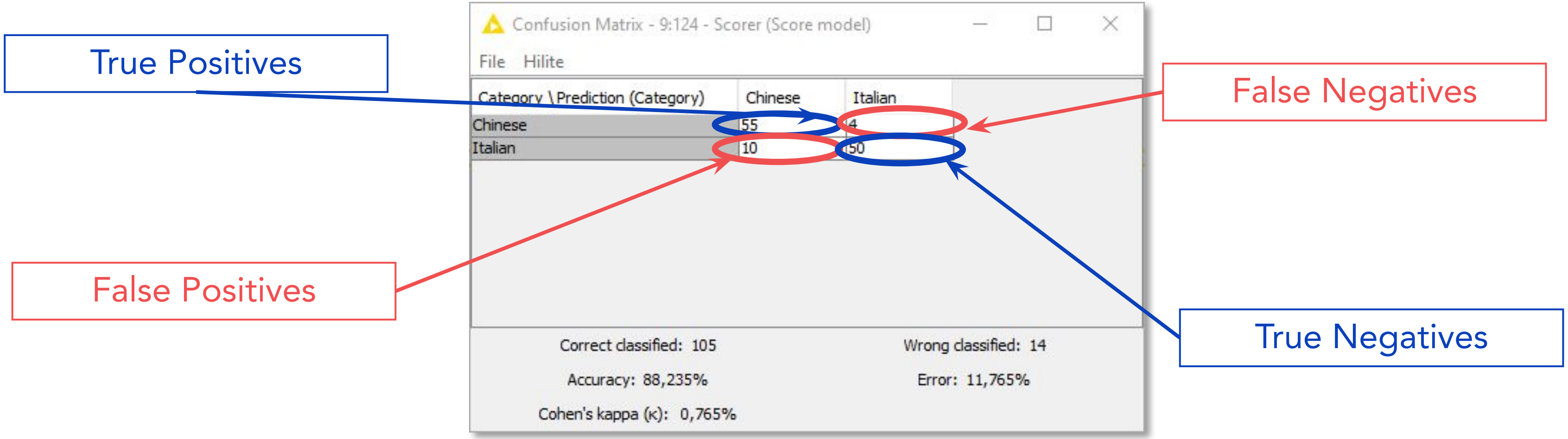
- Compare predicted results to known truth to evaluate model quality
- Confusion matrix shows the distribution of model errors
- An accuracy statistics table provides additional info





These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Scorer: Confusion Matrix



These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Machine Learning Concepts for NLP

- Introduction to NLP Models
  - Understand, interpret, generate human language
- Word2Vec
  - Maps words to vectors based on context
  - Semantic meaning reflected in vector proximity
- Contextualized Word Embeddings
  - Dynamic representation based on context
  - Example: "bank" in "river bank" vs. "bank account"

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Transformer Models and Their Evolution

- Introduction to Transformers
  - New class of models from 2018
  - Handle diverse languages and syntax
- BERT and GPT
  - BERT: Text classification and NER
  - GPT: Text generation
- Evolution of GPT
  - Improved accuracy and generation capabilities
- Which is better?

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

What Are Large Language Models (LLMs)?

- The adjective *large* refers to the *billions* of trainable parameters.
- Precursors to LLMs emerged in 2018 with models like BERT and the first Generative Pre-Trained model (GPT)
- Improved due to the following trends:
  - Increasing size (no. of trainable parameters)
  - Increased amount of data
  - Various fine-tuning methods
  - Architectural improvements
- A general trend: the larger, the better
  - OpenAI's GPT-4 (~1.76T) > GPT-3 (175B)
  - Other optimisations still yielding improved models (not only increased parameter counts).

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

How do LLMs "think"?

- LLMs function like highly sophisticated auto-completion systems (like text suggestions on smartphones).
- Fundamentally, LLMs are trained to suggest the most likely next word/token based on exposure large amounts of data.
- Every generated word is selected from an inventory of words; the LLM's vocabulary.
- Every time a word (or technically a 'token') is generated, it assigns a probability to every possible next word/token, and we usually pick the word with the highest probability.

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

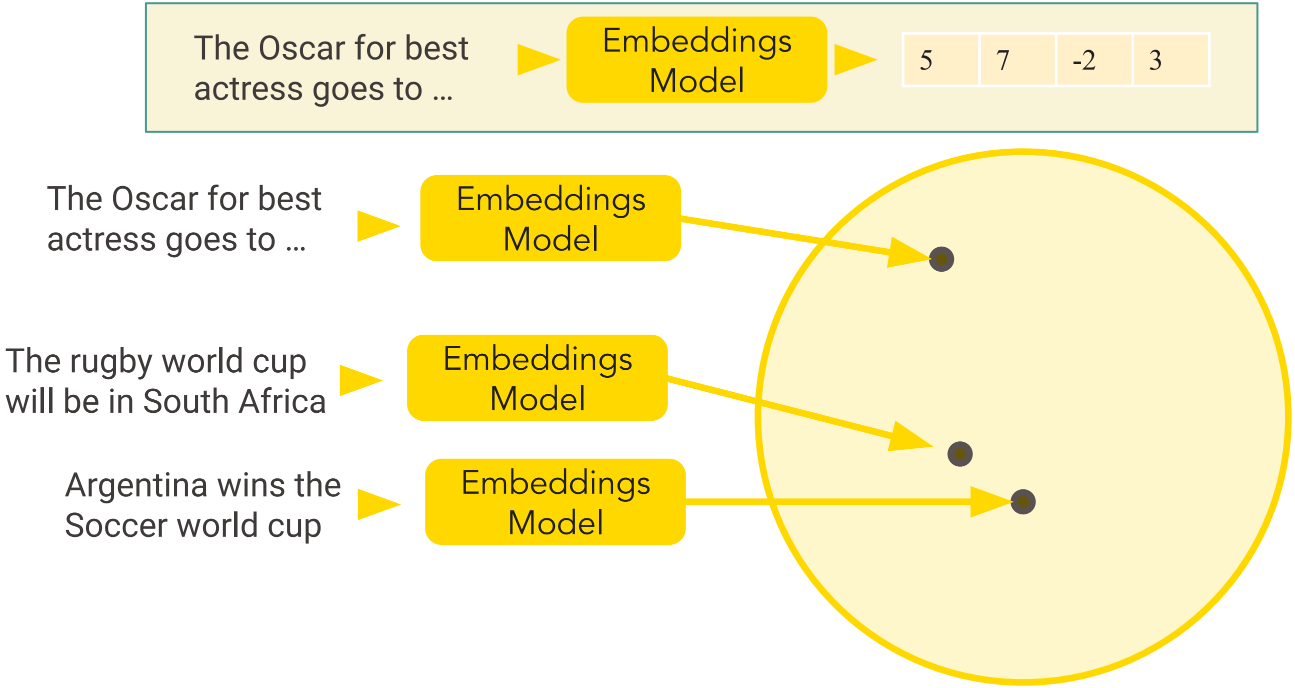
Embeddings

- Embeddings are extracted from the hidden (internal) state of an LLM.
- Under the hood, each word is represented by an n-dimensional vector called an 'embedding'.
- This is simply a list of n numbers, where the size 'n' just depends on the model itself.
- An embedding represents the semantic meaning of a word or phrase.
- Words with similar meaning will have similar embeddings.
- Another angle: Embeddings that are close together correspond to words with similar meaning.

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

What Do Embeddings Really Look Like?

- The different phrases or individual words can be represented in the ‘embedding space’
- The embedding for each word/phrase is a set of coordinates in this multidimensional space

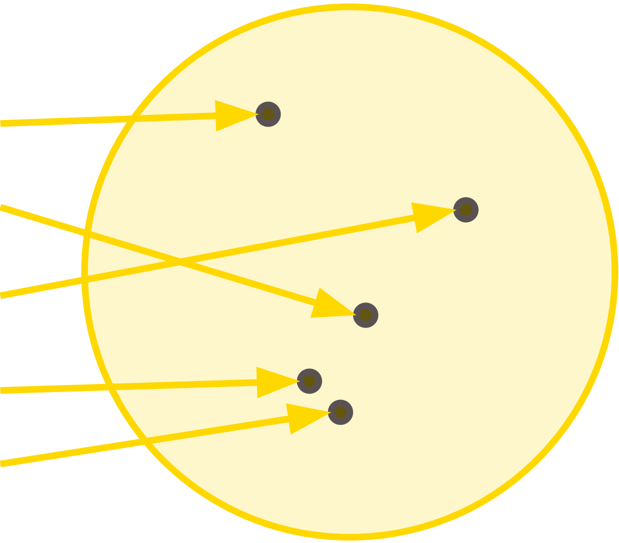


These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

What is a Vector Store?

- A database that stores texts/documents with their corresponding embeddings
- Easy to perform similarity search
- Texts with similar meanings have similar or close embeddings

Document	Embedding
The 2023 Oscar for best actress goes to …	5, 7, -2, 3
Argentina wins the soccer world cup	4, 6, -2, 4
The Kansas City Chiefs win the superbowl in 2023	4, 6, 5, -1
Bayern Munich wins the Bundesliga in 2023, again…	0, 4, 8, 9
VfB Stuttgart stays first class	0, 1, 9, 8



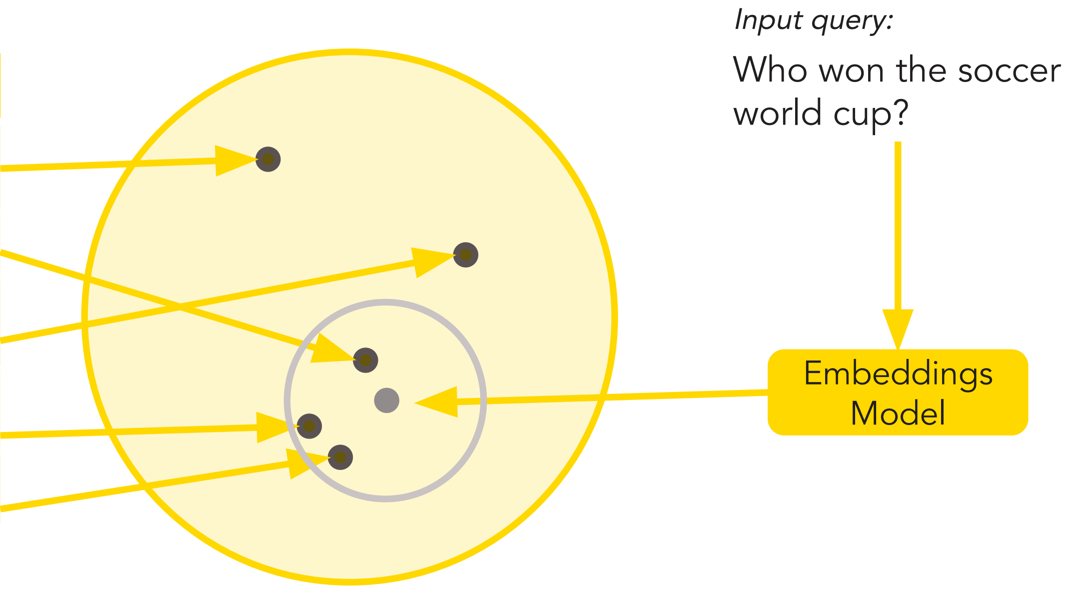
These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

How can we use it for Semantic (Similarity) Search?

- We have already indexed the embeddings vectors for some documents
- When a new document arrives, we get the embedding for this (using the LLM)
- We can then look for the embeddings in our index closest to the new document
- We can then retrieve the document associated with 'nearby documents'

Previously indexed documents and embeddings:

Document	Embedding
The 2023 Oscar for best actress goes to ...	5, 7, -2, 3
Argentina wins the soccer world cup	4, 6, -2, 4
The Kansas City Chiefs win the superbowl in 2023	4, 6, 5, -1
Bayern Munich wins the Bundesliga in 2023, again...	0, 4, 8, 9
VfB Stuttgart stays first class	0, 1, 9, 8



These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Practical Demonstration Using LLM

- LLM for Fraud Detection
  - Use embeddings from LLM for analysis
  - Detect fraud in Enron email dataset
- Text-Generation Capabilities
  - Flag suspicious activities without complex modelling
- KNIME Demo
  - Practical implementation and hands-on activity

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0

Knime Courses Access

- Code for free access to:
  - Online Courses
  - Certifications
  - Books
- Code: EC-BRUSSELS-24

These slides are a derivative of *KNIME Course Material* of KNIME AG used under CC BY 4.0