MACHINE LEARNING OF SIGNIFICANT TERMS

LANCASTER UNIVERSITY, SCC460, GROUP EPSILON

INFORMED SOLUTIONS (IS)

- Specialise in digital transformation, artificial intelligence, and data analytics
- Solve complex business problems
- Involved in Planning and Land Management, Healthcare, Energy, and Government

MOTIVATION

Currently - for a document set:

- IS system sorts/categorises documents based on significant terms
- IS data pre-processing removes non-English terms
- Problem - many business specific terms are removed, e.g. 'cil'

Ideally:

- Don't discard non-English terms
- Use NLP techniques that don't require a specific language
- Find significant non-English terms
- Then add them to the IS dictionary – keep words

RESEARCH QUESTIONS

- Conduct a comparison of techniques to find the significance of non-English terms:
    1. What NLP techniques can be used to find significant terms?
    2. Can a technique be used for terms not in a dictionary? Can they be adapted?
    3. What are the advantages and disadvantages of each short-listed technique?
    4. What is the performance like on an unstructured dataset?

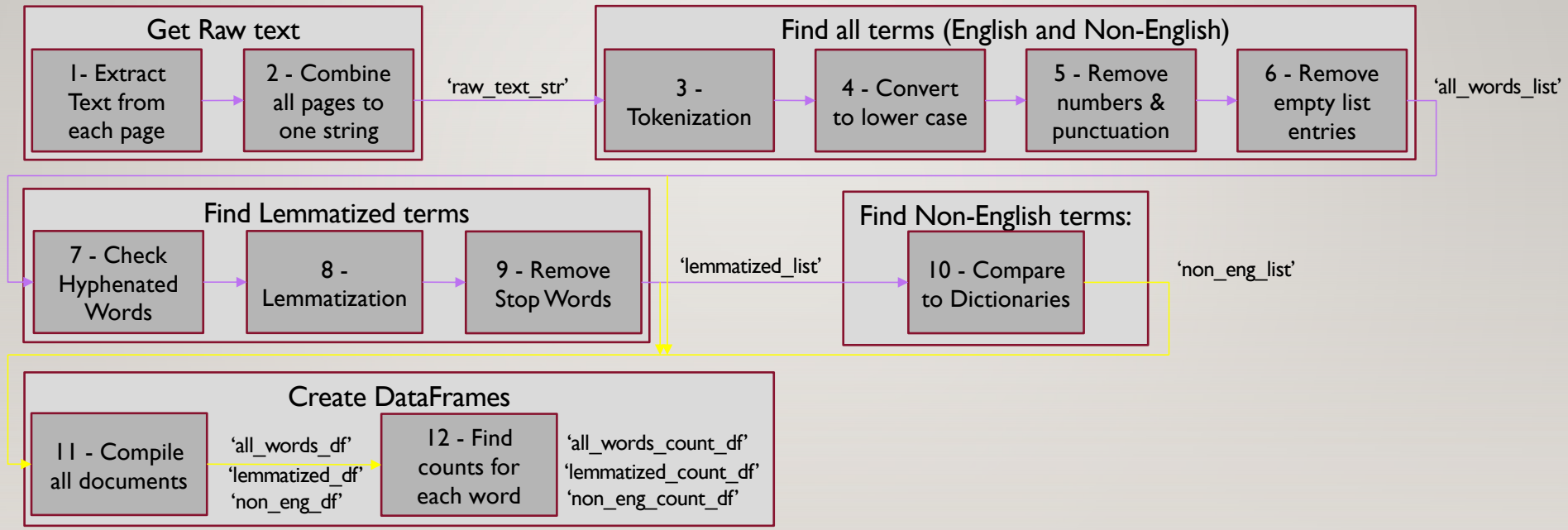
- Provide a Machine Learning Pipeline for the recommended technique

DESCRIPTION OF THE DATASET

OVERVIEW OF THE PROCESS

- Pre-processing - from documents to word lists
- Formatting – Word lists to useable data (DataFrames and counts)
- Generating significant terms for each technique
- Compare results
- Classification

FLOW CHART OF PRE-PROCESSING



PRE-PROCESSING

STEP 1:

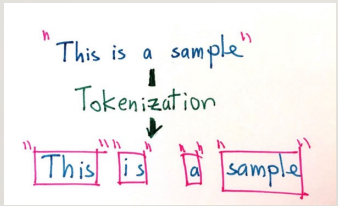
EXTRACT TEXT

- Currently only processing PDFs
- Will implement optical character recognition

STEP 3:

TOKENIZATION

- Splitting body of text into words



STEP 4:

CONVERT TO LOWER CASE

- Dictionaries only have lowercase
- Different case, different symbol

PRE-PROCESSING

STEP 5:

REMOVE SPECIAL CHARS

- Creates challenges to further processing
- Special characters and numbers are irrelevant
- Most models do not consider punctuation

STEP 7:

CHECK HYPHENATED WORDS

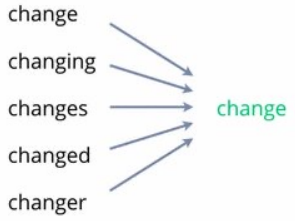
- Split up hyphenated words into individual words
- E.g. short-term, full-scale

PRE-PROCESSING

STEP 8:

LEMMATIZATION

- Converts words to its base form
- Multiple forms of a word causes noise and inaccuracy



STEP 9:

REMOVE STOP WORDS

- Set of commonly used words in a language
- Unimportant words
- Removal allows focus on important words instead

Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read

STEP 10:

COMPARE TO DICTIONARY

- Distinguish between English and Non- English words

FORMATTING

STEP 12:

FIND COUNTS FOR EACH WORD

- Calculate how many instances of each word are in each document, and in the corpus.
- Directly gives us 'bag of words'

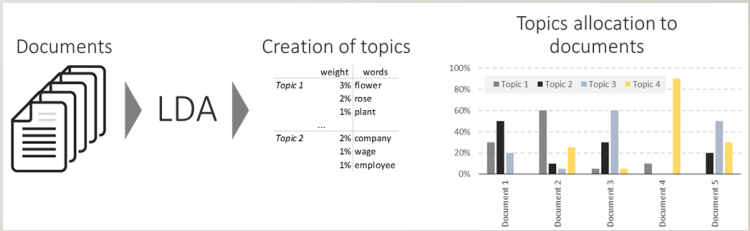
NLP TECHNIQUES

BAG OF WORDS

- Used to extract features from text
- Represents each document/string of text as a bag
- Disregards the grammar and gives out the frequency/count of every feature
- Discards structure and order of the words in a document

LATENT DIRICHLET ALLOCATION (LDA)

- Used for topics extraction from a corpus of documents
- A topic is represented as a weighted list of words
- Fast to run and intuitive – basically, bag of words with weights
- Major drawback is it requires a lot of fine tuning



TEXT RANK

- Graph-based algorithm used for keyword extraction and text summarization
- Based on Google's PageRank
- Each word is a node
- Measures the relationship between two or more words
- Calculates the weight for each word

OTHER TECHNIQUES CONSIDERED

TF-IDF

- Used for document similarity, not corpus term significance
- Can be adapted, but not ideal

BERT

- Relies on semantics of words
- Can't easily get semantics of non-English

HDP

- Similar to LDA
- Implementation complexity

RESULTS & COMPARISON

Bag of words

- Instant execution
- Human approach
- No deep analysis

BAG OF WORDS:

	words	numFound
0	altrincham	296
1	cil	185
2	centre	167
3	●	132
4	©	115
5	ltd	94
6	wa14	94
7	trafford	89
8	bowdon	82
9	rwp	72
10	hha	62
11	db	58
12	ful	51
13	wa15	50
14	mm	43

LDA

- Fast execution
- Actual ML model (can be improved with training)
- Needs hand tuning
- Non-deterministic

GENSIM (with 20 topics, 20 keywords):
NOTE: This list changes every execution

	words	numFound
0	rooflight	43
1	rwp	72
2	cil	185
3	sqm	25
4	johnwoodarchitect.co.uk	14
5	denotes	16
6	bowdon	82
7	21st	41
8	a1	33
9	●	14
10	hha	62
11	mh	41
12	altrincham	296
13	relates	26
14	ltd	94

TextRank

- Very slow execution
- Actual ML model (easy to train)
- Results sometimes unusual
- Close to Bag-of-words

TEXTRANK:

	words	numFound
0	altrincham	296
1	mm	43
2	wa14	94
3	trafford	89
4	bowdon	82
5	rwp	72
6	altrincham	296
7	ltd	94
8	a1	33
9	cil	185
10	ltd	94
11	ppe	15
12	rooflight	43
13	mh	41
14	wa15	50

FUTURE WORKS

- Classification:
  - Can we sort documents based on keywords?
    - E.g. LDA 'topics'
- Validation:
  - Request a list of top words expected from IS
  - For classification:
    - Explore existing system
    - Does our non-English data improve their system?