Assignment 2: Keyword Extraction

Mirudula E., MS18194
Akshay Shankar, MS18117
(Equal contribution)

1 Introduction

In this assignment, we will be utilizing TextRank and TF-IDF methods (via various python packages), to rank and extract the top ten keywords from 5 randomly chosen articles. We will then compare the results to the keywords provided by the authors and draw conclusions about the efficacy of these methods.

1.1 Articles to be used

- *Synthetic biology devices for in vitro and in vivo diagnostics*
- *Generation of Pseudomonas putida KT2440 Strains with Efficient Utilization of Xylose and Galactose via Adaptive Laboratory Evolution*
- *De Novo Designed Protein-Interaction Modules for In-Cell Applications*
- *Gauge Functions in Classical Mechanics: From Undriven to Driven Dynamical Systems*
- *The Merger Rate of Black Holes in a Primordial Black Hole Cluster*

1.2 Extracting usable text from the articles

```
from bs4 import BeautifulSoup
import requests
import re

def getText(url):
  p=requests.get(url) #get the html content of base-url page
  s=BeautifulSoup(p.content, "html.parser") #parse the first page
  a = s.find("div", {"class": "html-body"}) #retrieve article text

  #cleaning the text

  x = re.sub("\n", " ", a.text)
  x = re.sub("[^a-zA-Z.-]", " ", x)
  x = re.sub(" +", " ", x)
  x = re.sub("(\\b[A-Za-z] \\b|\\b [A-Za-z]\\b)", " ", x)
  x = re.sub(" +", " ", x)

  return x
```

The above code was used to scrape and clean the text from some of the articles (using BeautifulSoup + Regex). Few errors from this result were then manually corrected.

1.3 Extracting keywords

1.3.1 Gensim and Summa packages

In these cases, we simply have to use a single function from the packages like shown below.

```
from gensim.summarization import keywords as gkw
from summa import keywords as skw

def getKeywordsGensim(text, nwords = 10, lem = True):
  return gkw(text, words = nwords, split = True, lemmatize = lem)

def getKeywordsSumma(text, nwords = 10):
  return skw.keywords(text, words = nwords).split()
```

1.3.2 TF-IDF using NLTK

In this case, we construct the TF-IDF algorithm from scratch.

```
from operator import itemgetter
from nltk import tokenize
import nltk, math

nltk.download("stopwords")
nltk.download("punkt")

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

stop_words = set(stopwords.words('english'))

def getKeywordsNLTK(doc, nwords):
  total_words = doc.split()
  total_word_length = len(total_words)

  total_sentences = tokenize.sent_tokenize(doc)
  total_sent_len = len(total_sentences)

  tf_score = {}
  for each_word in total_words:
      each_word = each_word.replace('.','')
      if each_word not in stop_words:
          if each_word in tf_score:
              tf_score[each_word] += 1
          else:
              tf_score[each_word] = 1

  # Dividing by total_word_length for each dictionary element
  tf_score.update((x, y/int(total_word_length)) for x, y in tf_score.items())

  def check_sent(word, sentences):
    final = [all([w in x for w in word]) for x in sentences]
    sent_len = [sentences[i] for i in range(0, len(final)) if final[i]]
    return int(len(sent_len))

  idf_score = {}
  for each_word in total_words:
      each_word = each_word.replace('.','')
      if each_word not in stop_words:
          if each_word in idf_score:
              idf_score[each_word] = check_sent(each_word, total_sentences)
```

```
            else:
                idf_score[each_word] = 1

    # Performing a log and divide
    idf_score.update((x, math.log(int(total_sent_len)/y)) for x, y in idf_score
      .items())

    tf_idf_score = {key: tf_score[key] * idf_score.get(key, 0) for key in
      tf_score.keys()}

    def get_top_n(dict_elem, n):
      result = dict(sorted(dict_elem.items(), key = itemgetter(1), reverse =
      True)[:n])
      return result

    return get_top_n(tf_idf_score, nwords)
```

2 Results and Observations

We have extracted keywords from both the abstract as well as the main article of the 5 papers mentioned above. We have also compared against (non)-lemmatized extraction of Gensim to note the differences. There are some general remarks common to all these results:

- The Gensim and Summa results are nearly identical, as they both use the TextRank algorithm to extract keywords. Therefore, we will use "TextRank" to refer to both these packages.
- There is an option to "lemmatize" in Gensim, which eliminates duplicates/variants of the same words, such as "physical", "physically", "physics", etc. This gives better results.
- The NLTK/TF-IDF method performs poorly in case of the biology articles, and in general, it identifies more **frequently occurring** words rather than **important** ones.

In the following sections, we have displayed the results of our keyword extractions from the abstract and the article, with a detailed analysis of the accuracy of various packages.

Note: The rank of the resulting keywords decreases as we go down the list.

2.1 Article 1: Synthetic biology devices for in vitro and in vivo diagnostics

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NTLK	summa
Article: Synthetic biology devices for in vitro and in vivo diagnostics				
synthetic biology	diagnostics	diagnostic	Synthetic	growing need
diagnostics	diagnostic	synthetic	There	diagnostics
biosensing	synthetic	rapidly	biologists	diagnostic
synthetic gene networks	rapidly	new	devices	current efforts
nanobiotechnology	new	biology	enhance	goal offering promising results
	engineering	offering promising	environmental	synthetic
	biology	pathological	growing	rapidly
	offering promising	engineered gene	medical	real surveillance
	pathological	near	need	provide near
	engineered gene	real	vitro	technologies
	near	palette creating dynamic		equipment
	real	emerging		laboratory
	palette creating dynamic	confront		biology
	confront	diseases		untethering reactions
	emerging	circuits bring		antibody platforms
	diseases	equipment		pathological
	circuits bring	laboratory		new
	growing need	exquisitely		engineering
	efforts	cycles		engineered
	exquisitely	sensitive		sensitive

(a) Abstract

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NLTK	summa
Article: Synthetic biology devices for in vitro and in vivo diagnostics				
synthetic biology	based	based	(v/v)	based
diagnostics	synthetic	synthetically	CC-Tet-A2B2	synthetic
biosensing	synthetically	cell	DNA	synthetically
synthetic gene networks	cells	diagnostic	DNA-binding	cells
nanobiotechnology	cell	engineer	However	cell
	diagnostic	mas	LacI*	diagnostics
	engineering	genes	PPIs	diagnostic
	engineered	included	ParaBAD	engineering
	engineer	reporters	SUMO	engineered
	ma	circuit	The	engineer
	mas	bacterial	mM	gene
	gene	nature	sequences	genes
	genes	sensors	C	include
	include	like	uL	including
	including	sequence		included
	included	targeting		report
	report	bacteria		reporting
	reporting	application		reporter
	reporter	activation		reporters
	reporters	vivo		rna

(b) Main article

We note that the TextRank keywords extracted from the abstract and entire article are somewhat similar to the author-provided ones. Though we don't see the exact matching of author-provided keywords and the keywords captured by TextRank, we see that the keywords such as "diagnostics", "synthetic", "biology", "gene" are also part of author-provided keywords. However, we note that the author-provided keyword "synthetic biology" was captured as two separate keywords "synthetic" and "biology" which might be because there are two author-provided keywords, "synthetic biology" and "synthetic gene network", that contain the same word "synthetic". Similarly a part of "synthetic gene network" was also captured by TextRank as "synthetic" and "gene". The author-provided keyword "biosensing" is somewhat captured by Gensim (from the entire article keyword extraction) as "sensors". However, we also note that TextRank has extracted several extra words that may be considered "common" in biology, and thereby are not useful as keywords.

Now, if we note the keywords that were extracted using NLTK that uses TF-IDF method, we don't see any keywords matching with that to the author-provided keywords. This method seems to work poorly as we know that in general all biology articles do contain the words like "(v/v)", "uL", "sequence", "mm" in a higher frequency at least in their methods section. Since, TF-IDF tries to rank the words as keywords primarily based on their frequency, it is no wonder why NLTK extracted those words as keywords.

Besides, the frequency of the author-provided keyword, "nanobiotechnology" in the entire article is just once and this might be a possible reason why Gensim, Summa and NLTK did not capture that keyword. The authors must have provided that keyword just to highlight the discipline of their work.

2.2 Article 2: Generation of Pseudomonas putida KT2440 Strains with Efficient Utilization of Xylose and Galactose via Adaptive Laboratory Evolution

Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NTLK	summa
onas putida KT2440 Strains with Efficient Utilization of Xylose and Galactose via Adaptive Laboratory Evolution			
strains	strains	2-ketogluconate	encoding
encoding xylonate	encoding xylonate	efficient	strains
strain growth	sugar	evolution	strain growth
sugars	biomass	generation	sugars
sugar	product	KT2440	sugar
biomass	growth	strains	biomass
product	optimized	utilization	product
optimized	galactose	xylose	kt
utilize	sequencing transcriptomic	g/L	optimize engineered
utilized	engineered	h-1	utilize
indigoidine production	heterologous		utilized
optimize engineered	indigoidine		indigoidine production
galactose	evolution		optimized
sequencing transcriptomic	efficient		galactose
heterologous	uridylyltransferase		sequencing transcriptomic
evolution	ketogluconate operon		produced
efficient	respectively		produce
uridylyltransferase	platform		g
ketogluconate operon	revealed significant		heterologous
respectively	laboratory		evolution

(a) Abstract

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NLTK	summa		
Article: Generation of Pseudomonas putida KT2440 Strains with Efficient Utilization of Xylose and Galactose via Adaptive Laboratory Evolution						
adaptive laboratory evolution	mutations	mutated	A6_F90_I1	strains		
Pseudomonas putida	mutational	strain	ALE	strain		
xylose	mutation	growth	DNA	mutations		
galactose	mutated	gene	In	mutational		
Weimberg pathway	strains	cells	KT2440	mutation		
Leloir pathway	strain	sugar	P	mutated		
	cell growth	utilized	The	cell growth		
	genes	galactose	USA	genes		
	gene	dna	Weimberg	gene		
	cells	xylose	galETKM	cells		
	sugars	pathway	h-1	galactose		
	sugar	regions	utilization	sugars		
	utilization	encode	xylD	sugar		
	utilizing	biomass	xylose	xylose		
	utilize	ale		utilization		
	utilized	study		utilizing		
	galactose	glucose		utilize		
	dna	protein		utilized		
	xylose	changing		dna		
	pathways	improves		region		

(b) Main article

We note that the keywords such as “galactose”, “evolution” extracted from the abstract by TextRank matches/partly matches with the author-provided ones. The keywords “xylose”, “galactose” and “pathways” extracted from the article by TF-IDF method also matches/partly matches with the author-provided ones. Besides, “indigoidine” captured by the TextRank method will be a good suggestion to be added as the keyword. However, we also note that issues with the TextRank method observed in case of keyword extraction from article 1 exists here also.

Similarly we note the keywords such as “xylose”, “evolution” extracted from the abstract by TF-IDF method also matches/partly matches with the author-provided ones. The keywords “xylose”, “Weimberg” extracted from the article by TF-IDF method also matches/partly matches with the author-provided ones. This method might have partly captured the author-provided keyword “Weimberg pathway” because in most places in the article we find “Weimberg and Leloir pathway” and also the word “pathway” is succeeded after other words such as “production pathway”, “utilization pathway”, “galactose oxidative pathway” etc in addition to “Weimberg”.

In addition we also note that the NLTK keyword “KT2440” (extracted from article and abstract) is actually the strain detail of the species “Pseudomonas putida”, which is one of the author-provided keywords. And also in the article they have mentioned that “Pseudomonas putida KT2440” will be hereafter mentioned as “KT2440”. Similarly, in the article we see that they refer to “adaptive laboratory evolution” (author-provided keyword) as “ALE” and this is also captured by NLTK and Gensim (extracted from the article). Thus, we can say that the TF-IDF method of keyword extraction is somewhat better than the TextRank method of keyword extraction with respect to the number of keywords in common (also considering the ranking order of those keywords) between the author-provided keywords and the ones that were extracted. However, we see that the TF-IDF method still extracted “g/L” and “h-1” as the keywords due to their high frequency in the article.

2.3 Article 3: De Novo Designed Protein-Interaction Modules for In-Cell Applications

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NTLK	summa
Article: De Novo Designed Protein-Interaction Modules for In-Cell Applications				
de novo protein design	designed	designs	Applications	designed
coiled coil	design	coil	De	design
transcriptional control	designs	controlling	Designed	designs
Lac repressor	coiled coils	ways	In-Cell	coiled coils
artificial transcription factor	coil	proteins	Lac	coil
	controlled	coli	Modules	ways
	controlling	different	Novo	controlled
	ways	interaction	Protein-Interaction	controlling
	proteins	parallel	The	coli
	protein interactions control	repression assay	protein-protein	different
	different	homotetramer		repression assay
	coli	natural biological		homotetramer
	interaction	highly		parallel
	repression assay	modifications		biology
	parallel	specific		protein interactions control
	homotetramer	components		natural biological
	natural biological	finer		highly
	highly	environment		modification
	modification	cellular		modifications
	modifications	considerable potential		specificity

(a) Abstract

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NLTK	summa
Article: De Novo Designed Protein-Interaction Modules for In-Cell Applications				
de novo protein design	peptides	peptides	DNA	cc
coiled coil	peptide	proteins	For	peptides
transcriptional control	protein	transcriptional	However	peptide
Lac repressor	proteins	ccs	In	protein
artificial transcription factor	transcription	designs	One	proteins
	transcriptional	dna	RBS	transcription
	ccs	laci	RNA	transcriptional
	designed	domain	Synthetic	mm
	design	cells	The	laci
	designing	ppi	These	cells
	designable	sumo	This	cell
	designs	residuals	Using	ccs
	dna	plasmids	expression	designed
	laci	controller	luxCDABE	design
	domain	naturally	networks	designing
	cells	interacted		designable
	cell	application		designs
	ppi domains	dimerize		domains
	ppis	acid		domain
	sumo	repressed		dna

(b) Main article

The keywords “coiled coils”, “protein”, “design” extracted from the abstract using TextRank method partly match with the author-provided keywords “coiled coil” and “de novo protein design”. Similarly, the keywords “protein”, “design”, “transcriptional”, “lacI”, “ccs” extracted by the TextRank method match quite well with the author-provided keywords “de novo protein design”, “transcriptional control”, “Lac repressor”, and “coiled coil”. It is important to note that LacI is the gene for the Lac repressor and thus, we can say that the authors might have mentioned the protein, Lac repressor instead of LacI, as they both exactly mean the same indirectly. Also, in the article, the authors have referred to “coiled coil” as “ccs” and this is well captured by the TextRank method.

The keywords “De”, “Novo”, “Designed”, “Lac” extracted from the abstract using NLTK- TF-IDF method partly match with the author-provided keywords “de novo protein design” and “Lac repressor”. The keyword “protein-protein interaction” captured by this method (from abstract) could be a good suggestion that is missing in the author’s list of keywords. However, the keywords extracted from the article using this method is not very efficient and mostly identifies unwanted words that occur at high frequency in the article such as “DNA”, “RNA”. One must note that the suggested keyword “protein-protein interaction” is not captured while extracting the keywords from the article might be because in most places in the article it is abbreviated as “PPI”.

It is important to note that the author-provided keyword “artificial transcription factor” is not captured by either of these methods, maybe because this word just occurs once in the article. Authors might have included this keyword as their current work can also be used to design “artificial transcription factors”. In conclusion, although both TextRank and TF-IDF methods have their own drawbacks as discussed earlier, with respect to article 3, we could say that the TextRank method performed more efficiently than the TF-IDF method in keyword extraction.

2.4 Article 4: Gauge Functions in Classical Mechanics: From Undriven to Driven Dynamical Systems

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NLTK	summa
Article: Gauge Functions in Classical Mechanics: From Undriven to Driven Dynamical Systems				
Lagrangian formalism	novel gauge functions	novel	Classical	novel gauge functions
null Lagrangians	energy function	energy function	Driven	energy function
gauge functions	quantum field	quantum field	Dynamical	quantum field
forces	gauges	gauges	From	gauges
classical mechanics	forces	define	Functions	obtained results
	undriven physical	undriven physical	Gauge	forces
	obtained results	forces	Mechanics	undriven physical
	define	obtained results	Systems	define
	directly affect	directly affect	Undriven	directly affect
	non classical mechanics	non classical mechanics		non classical mechanics
	theories	theories		theories
	phenomenon	phenomenon		phenomenon

(a) Abstract

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NLTK	summa
Article: Gauge Functions in Classical Mechanics: From Undriven to Driven Dynamical Systems				
Lagrangian formalism	function	functional	CM	functions
null Lagrangians	functional	lagrangians	Equation	functions
gauge functions	lagrangian	gauges	Galilean	functional
forces	lagrangian	generate	Helmhotz	lagrangian
classical mechanics	gauge	constants	Lagrangian	lagrangians
	gauges	equation	Lagrangians	gauge
	general	force	NLs	gauges
	generalize	differences	ODEs	general
	generalized	physics	SLs	generalize
	generated	energy	equation	generalized
	generate			forces
	constant			force
	constants			constant
	equations			constants
	equations			equations
	forces			equation
	forces			different
	different			difference
	difference			differently
	differently			differences
	differences			physical
	physical			physically
	physically			physics
	physics			energy

(b) Main article

The TextRank keywords from the abstract are quite similar to the author-provided ones, however, it also has several extra words that may be considered "common" in physics, and thereby are not useful as keywords (such as "equations", "constants", "generate", "energy", "differences", etc). From the entire article, we see that the main keywords such as "lagrangian", "force", "gauges" are captured by TextRank, and even some suggestions like "functional" might be good keywords but are missing from the author's list.

The NLTK/TF-IDF method works quite well both from the abstract and the article by identifying "gauge functions", "classical mechanics", and "lagrangian", which match with the author keywords, although these were recognized as two separate keywords instead of phrases. It also captures some useful words like "dynamical systems", "Galilean", "Helmhotz", and "ODEs" which were absent from the author provided keywords.

So in this case, TF-IDF does a better job except that it captured individual words instead of phrases. The keyword "null lagrangian" however has not been captured, and this might be because its occurrence is a low 22 times, as compared to other keywords which occurred in the order of $\approx$ 100 times in the article.

2.5 Article 5: The Merger Rate of Black Holes in a Primordial Black Hole Cluster

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NTLK	summa
Article: The Merger Rate of Black Holes in a Primordial Black Hole Cluster				
black hole cluster	black	black	PBHs	changed
black hole merging	hole	hole	In	merger
merger rate	mass spectrum	mass spectrum	The	rate
primordial black holes	rate	rate	characteristics	black
Fokker-Planck equation	changed	changed	close	holes
	central	central	globular	holes
	globular star	globular star	investigated	significantly
	clusters	clusters	paper	central
			primordial	characteristics
			typical	close

(a) Abstract

Authors-Keywords	Gensim 3.8.0	Gensim 3.8.0 (Lemmatize=True)	NLTK	summa
Article: The Merger Rate of Black Holes in a Primordial Black Hole Cluster.				
black hole cluster	mass	masses	CBH	mass
black hole merging	masses	cluster	Equation	masses
merger rate	clusters	merger	LIGO	clusters
primordial black holes	clusters	black	PBH	cluster
Fokker-Planck equation	merger	gravitational	PBHs	merger
	black hole mergers	pbh	Universe	black
	gravitational	cbh	Virgo	hole
	pbhs	time	black	merger
	pbh	holes	evolution	time
	cbh	density	merger	gravitational
	time			cbh
	holes			holes
	density			pbh
				density

(b) Main article

The TextRank keywords include "black", "holes", "merger", "cluster", "cbh" and "pbh" which all match the author provided keywords. The word "Primordial black hole" has been abbreviated to PBH in the article, resulting in only PBH being identified as a keyword. Some words are captured, but are too general for the field of astronomy, like "gravitational", "masses", "time" and "density", so these may not be very useful. Here too, we notice that "black" and "hole" are captured separately and not as a phrase. Also, the abstract produces some good keywords such as "mass spectrum" and "globular star" which were not found from the main article.

The NLTK/TF-IDF keywords have some good suggestions too, such as "PBH", "LIGO", "evolution" and "merger". But it also captures common words like "Universe" and "Virgo". The keywords from the abstract also have a mix of good words ("PBH", "globular", "primordial") in the midst of common words.

The author keyword "Fokker-Planck equation" only appears twice in the article, thereby it was not suggested by either TextRank or TF-IDF methods since it was too obscure in relation to other words. The authors must have included it, as it might be indicative of a niche field that is loosely related to the contents of the article.
