**SUMMARIZING WORDS USING LEXICAL ANALYSIS AND NLP**

**Abstract:**

This is an overview of Word Summarization. Word Summarization is a challenging problem these days. Due to the great amount of information we are provided with and thanks to the development of Internet technologies, needs of producing summaries have become more and more widespread. Summarization is a very interesting and useful task that gives support to many other tasks as well as it takes advantage of the techniques developed for related Natural Language Processing tasks.This paper may help us to have an idea of what Word Summarization is and how it can be useful for.

**Introduction:**

Word summarization is the technique for generating a concise and precise summary of voluminous texts while focusing on the sections that convey useful information, and without losing the overall meaning.Automatic word summarization aims to transform lengthy documents into shortened versions, something which could be difficult and costly to undertake if done manually.Machine learning algorithms can be trained to comprehend documents and identify the sections that convey important facts and information before producing the required summarized words. For example, the image below is of this news article that has been fed into a machine learning algorithm to generate a summary.

**Literature survey :**

Single Document Summarization: Various technical documents were focused in single-document summarization.Luhn in 1958 shows the significance of words based on frequency measures. He deleted the stop words and rest words are given a hierarchy starting from root and index describes the significance of each word.This is calculated on the number of occurrences in a document called as significant factor and are ranked. Based on ranking top sentences are selected to form a summary.

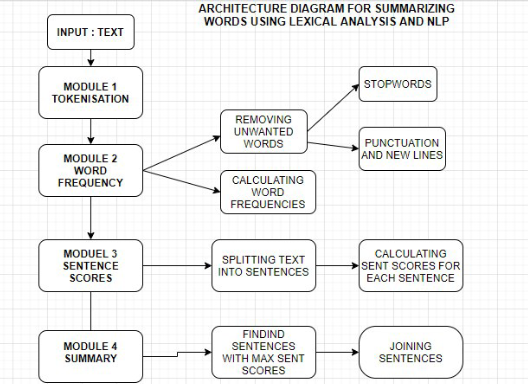
Baxendale in 1958 focused on sentence position to find the salient features. He took 200 paragraphs and examined that in 85% of paragraphs topic sentences are placed in the beginning while in rest 7% he found, it occurred in the last.

Edmundson in 1969 proposed a typical structure that produces extracts. In the beginning he took around 400 technical documents and build a protocol producing manual extracts. He addressed the above two features (word frequency, word position) and gave the two new features named cue words and skeleton (title or heading). Also the weights were attached with these. He evaluated and found that 44% machine extracts matched with manual extracts.

Various other pioneers were there who applied different techniques in single document summarization:

* In 1961 G.J. Rath used lexical indicators to determine the relevant information from documents.
* In 1995 Julian Kupiec used algebraic method to determine different features like uppercase words, length, position of words by using naïve-bayes classifier.
* In 1997 ChinYew Lin determine the position of sentences by using algebraic methods.
* In 1999 Eduard Hovy used symbolic word knowledge with strong NLP processing to show the concepts relevancy.
* In 2005 S.P Yong used neural netwok. He showed Summarization = Text pre-processing sub-system + Keywords Extraction sub-system + Summary production sub-system.
* In 1976 M.A. K. Halliday used lexical semantic relationships to build lexical cohesion blocks and their patterns.
* In 1984 Ruqaiya Hasan used lexical cohesion to identify similarity chains.
* In 1988 William C.Mann used RST (rhetorical structure theory) to encode the terminal nodes of a tree.
* In 1991 Jane Morris used cohesion chains to determine the sequence of associated words.
* In 1997 Branimir Boguraev used saliency-based content characterization to rank the important sentences in unstructured document.
* In 2010 Li Chengcheng used RST to analyze candidate sentence, discover rhetoric relations and give the construction.
* In 2000 Hongyan Jing used human abstraction concept by taking the closely realated sentences and eliminating the extra ones.

**Architecture diagram:**

****

**List of modules:**

* Tokenization
* Word Frequency
* Sentence scores
* Summarization

**Module 1:**

**Tokenization :**

Tokenization is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

* Sentences tokenization
* Words tokenization

Sentence tokenization - Splitting sentences in the paragraph.

Word tokenization - Splitting words in a sentence.

**Pseudocode:**

Tokenize the text into words:

Tokens = token.text **FOR** token **IN** doc

PRINT "Tokens"

**Module 2:**

**Word Frequency:**

Now calculate the weighted occurrence frequency of all the words.This is done after removing all stopwords and punctuation from the entire text. Also we need to find the normalized word frequencies of all words. To find the normalized frequencies,first of all the frequency of the most recurrent word of all the word frequencies (max\_wordfrequency) should be found. Then the count of each words are divided by max\_wordfrequency.This is done by the following formula

Normalized frequency of a word = Frequency of that word/Frequency of the most recurrent word.

**Pseudocode:**

Remove stopwords and punctuation

**FUNCTION** word\_frequencies

**FOR** word **IN** text

**FOR** word **NOT** **IN** stopwords

**FOR** word **NOT IN** punctuation

**IF** word **NOT IN** word\_frequencies\_keys()

word\_frequencies[word] = 1

**ELSE**

word\_frequencies[word] += word\_frequencies[word]

**ENDIF**

**ENDFOR**

**ENDFOR**

**ENDFOR**

**END FUNCTION**

Find max value of word\_frequencies\_keys()

**FUNCTION** Normalized\_word\_freq

**FOR** word **IN** word\_frequencies\_keys()

word\_frequencies[word] = word\_frequencies[word]/max\_word\_frequency

**ENDFOR**

**END FUNCTION**

**Module 3:**

**Sentence scores:**

Let's tokenize the text into sentences.

Let’s substitute each of the words found in the original sentences with their weighted frequencies. Then, we’ll compute their sum.

Since the weighted frequencies of the insignificant words, such as stop words and special characters, which were removed during the processing stage, is zero,so it’s not necessary to add them.

From the sum of the weighted frequencies of the words, we can deduce that the first sentence carries the most weight in the paragraph. Therefore, it can give the best representative summary of what the paragraph is about.

**Pseudocode :**

Tokenize the text into sentences

**FUNCTION** sentence\_scores

**FOR** sentence **IN** sentence\_tokens

**FOR** word **IN** sentence

**IF** word **IN** word\_frequencies\_keys()

**IF** sentence **NOT IN** sentence\_scores\_keys()

Sentence\_scores[sentence] = word\_frequencies[word]

**ELSE**

Sentence\_scores[sentence] += word\_frequencies[word]

**ENDIF**

**ENDIF**

**ENDFOR**

**ENDFOR**

**END FUNCTION**

**Module 4:**

**Summarization :**

This is the last step of word summarization. Here we are going to find the number of sentences to generate the best summary. So we need to know what percentage of the entire text we need and then multiply this percentage with number of sentence tokens that we already calculated.This is how the number of sentences for the summary can be calculated .And now we need to add those number of sentences with high scores for the summary . For example, let's consider If the first sentence is combined with the third sentence, which is the second-most weighty sentence in the paragraph, a better summary can be generated.

**Pseudocode:**

Select number of sentences to summarize in percentage //select\_length

Summary = nlargest(select\_length,sentence\_scores,key = sentence\_scores.get)

PRINT "Summary"

**Result Discussion:**

**Module 1:**

**Tokenization :**

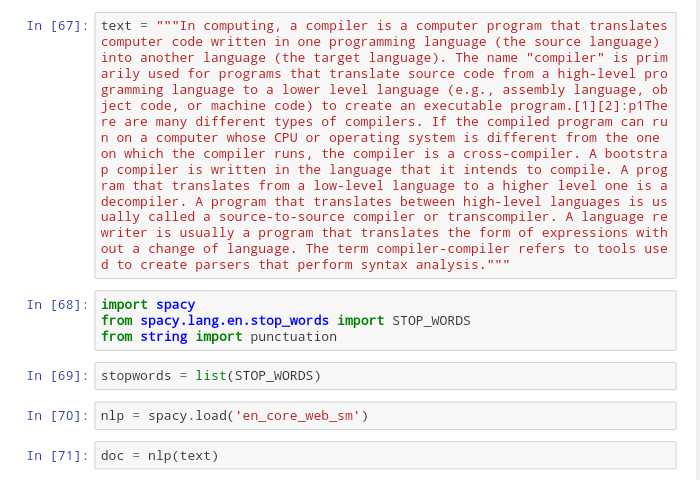
Let's take the following paragraph as an example.

text = """

In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language). The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program.[1][2]:p1

There are many different types of compilers. If the compiled program can run on a computer whose CPU or operating system is different from the one on which the compiler runs, the compiler is a cross-compiler. A bootstrap compiler is written in the language that it intends to compile. A program that translates from a low-level language to a higher level one is a decompiler. A program that translates between high-level languages is usually called a source-to-source compiler or transcompiler. A language rewriter is usually a program that translates the form of expressions without a change of language. The term compiler-compiler refers to tools used to create parsers that perform syntax analysis.

"""





Here a text is taken to summarize and the whole text is tokenized into words.

**Module 2:**

**Word Frequency:**





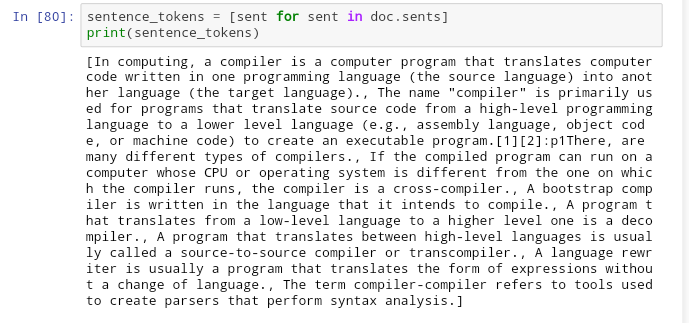
We are going to find word frequency. Before that the text should be cleaned i.e, removing the stopwords such as the, a, an, was, when,etc. and punctuation.But here we are not exactly cleaning the text. We are just not including the punctuation and stopwords and then we find the word frequencies for each word.

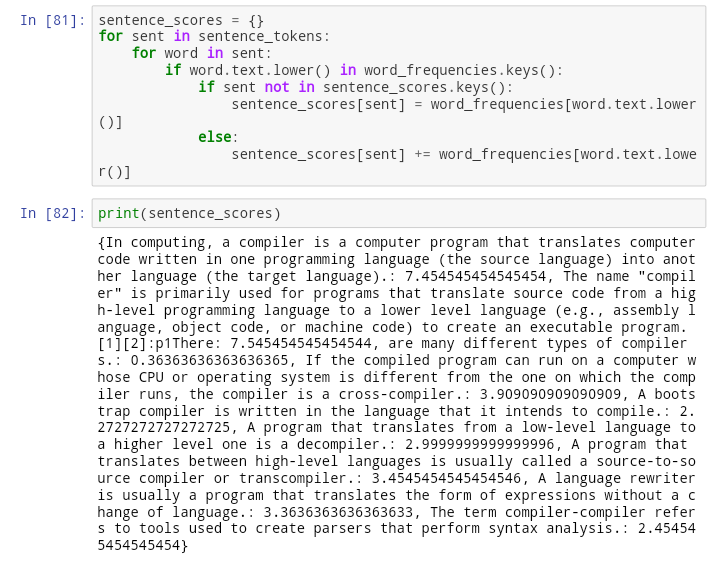
Then we find the frequency of the most recurrent word of all the word frequencies. Now normalized frequencies of all words can be calculated. This is done by the formula

Normalized frequency of a word = Frequency of that word/Frequency of the most recurrent word.

**Module 3:**

**Sentence scores :**

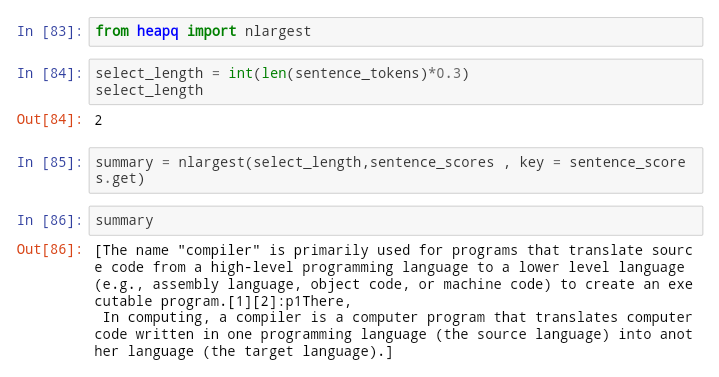


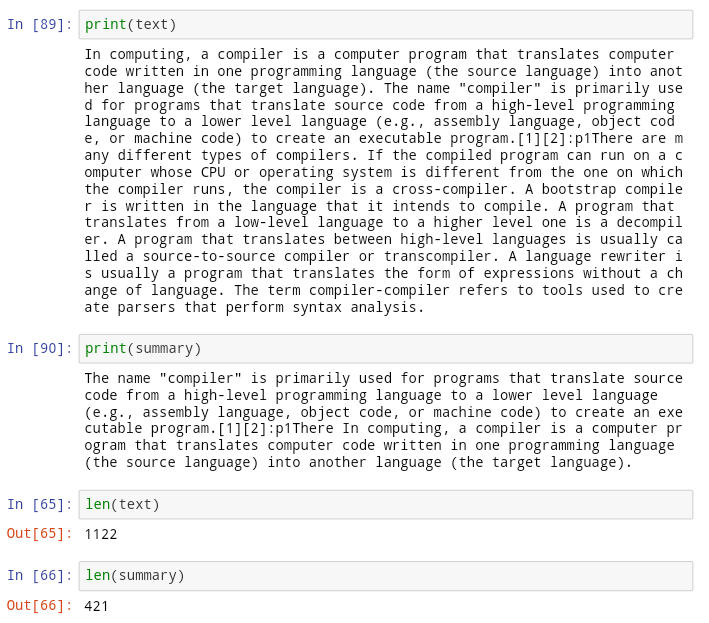
****

Here we tokenize the text into sentences. Then we substitute each of the words found in the original sentence with their weighted frequencies and their sum is computed. This is repeated for all the sentences and their sum will be computed.

**Module 4:**

**Summarization :**





Here we are going to find the number of sentences needed for the summarization(select\_length).

In the above example We are looking to get only 30 percent of the whole text. So we multiply 0.3 with the number of sentences of the whole text and we get

Only 2 sentences. Now we should add the 2 sentences with highest sentence scores in the sentence score list and thus a summary is generated.

**Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Input** | **Tokenization** | **Word frequency** | **Sentence scores** | **Summary** |
| 1 | A computer is a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming. Modern computers have the ability to follow generalized sets of operations, called programs. These programs enable computers to perform an extremely wide range of tasks. A "complete" computer including the hardware, the operating system (main software), and peripheral equipment required and used for "full" operation can be referred to as a computer system. This term may as well be used for a group of computers that are connected and work together, in particular a computer network or computer cluster. | ['A', 'computer', 'is', 'a', 'machine', 'that', 'can', 'be', 'instructed', 'to', 'carry', 'out', 'sequences', 'of', 'arithmetic', 'or', 'logical', 'operations', 'automatically', 'via', 'computer', 'programming', '.', 'Modern', 'computers', 'have', 'the', 'ability', 'to', 'follow', 'generalized', 'sets', 'of', 'operations', ',', 'called', 'programs', '.', 'These', 'programs', 'enable', 'computers', 'to', 'perform', 'an', 'extremely', 'wide', 'range', 'of', 'tasks', '.', 'A', '"', 'complete', '"', 'computer', 'including', 'the', 'hardware', ',', 'the', 'operating', 'system', '(', 'main', 'software', ')', ',', 'and', 'peripheral', 'equipment', 'required', 'and', 'used', 'for', '"', 'full', '"', 'operation', 'can', 'be', 'referred', 'to', 'as', 'a', 'computer', 'system', '.', 'This', 'term', 'may', 'as', 'well', 'be', 'used', 'for', 'a', 'group', 'of', 'computers', 'that', 'are', 'connected', 'and', 'work', 'together', ',', 'in', 'particular', 'a', 'computer', 'network', 'or', 'computer', 'cluster', '.', '\n'] | {'computer': 6, 'machine': 1, 'instructed': 1, 'carry': 1, 'sequences': 1, 'arithmetic': 1, 'logical': 1, 'operations': 2, 'automatically': 1, 'programming': 1, 'Modern': 1, 'computers': 3, 'ability': 1, 'follow': 1, 'generalized': 1, 'sets': 1, 'called': 1, 'programs': 2, 'enable': 1, 'perform': 1, 'extremely': 1, 'wide': 1, 'range': 1, 'tasks': 1, 'complete': 1, 'including': 1, 'hardware': 1, 'operating': 1, 'system': 2, 'main': 1, 'software': 1, 'peripheral': 1, 'equipment': 1, 'required': 1, 'operation': 1, 'referred': 1, 'term': 1, 'group': 1, 'connected': 1, 'work': 1, 'particular': 1, 'network': 1, 'cluster': 1} | {A computer is a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming.: 3.666666666666667, Modern computers have the ability to follow generalized sets of operations, called programs.: 1.9999999999999998, These programs enable computers to perform an extremely wide range of tasks.: 1.8333333333333337, A "complete" computer including the hardware, the operating system (main software), and peripheral equipment required and used for "full" operation can be referred to as a computer system.: 4.499999999999999, This term may as well be used for a group of computers that are connected and work together, in particular a computer network or computer cluster.  : 3.666666666666666} | A "complete" computer including the hardware, the operating system (main software), and peripheral equipment required and used for "full" operation can be referred to as a computer system. |
| 2 | 1.In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language). The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program.[1][2]:p1There are many different types of compilers. If the compiled program can run on a computer whose CPU or operating system is different from the one on which the compiler runs, the compiler is a cross-compiler. A bootstrap compiler is written in the language that it intends to compile. A program that translates from a low-level language to a higher level one is a decompiler. A program that translates between high-level languages is usually called a source-to-source compiler or transcompiler. A language rewriter is usually a program that translates the form of expressions without a change of language. The term compiler-compiler refers to tools used to create parsers that perform syntax analysis. | ['1.In', 'computing', ',', 'a', 'compiler', 'is', 'a', 'computer', 'program', 'that', 'translates', 'computer', 'code', 'written', 'in', 'one', 'programming', 'language', '(', 'the', 'source', 'language', ')', 'into', 'another', 'language', '(', 'the', 'target', 'language', ')', '.', 'The', 'name', '"', 'compiler', '"', 'is', 'primarily', 'used', 'for', 'programs', 'that', 'translate', 'source', 'code', 'from', 'a', 'high', '-', 'level', 'programming', 'language', 'to', 'a', 'lower', 'level', 'language', '(', 'e.g.', ',', 'assembly', 'language', ',', 'object', 'code', ',', 'or', 'machine', 'code', ')', 'to', 'create', 'an', 'executable', 'program.[1][2]:p1There', 'are', 'many', 'different', 'types', 'of', 'compilers', '.', 'If', 'the', 'compiled', 'program', 'can', 'run', 'on', 'a', 'computer', 'whose', 'CPU', 'or', 'operating', 'system', 'is', 'different', 'from', 'the', 'one', 'on', 'which', 'the', 'compiler', 'runs', ',', 'the', 'compiler', 'is', 'a', 'cross', '-', 'compiler', '.', 'A', 'bootstrap', 'compiler', 'is', 'written', 'in', 'the', 'language', 'that', 'it', 'intends', 'to', 'compile', '.', 'A', 'program', 'that', 'translates', 'from', 'a', 'low', '-', 'level', 'language', 'to', 'a', 'higher', 'level', 'one', 'is', 'a', 'decompiler', '.', 'A', 'program', 'that', 'translates', 'between', 'high', '-', 'level', 'languages', 'is', 'usually', 'called', 'a', 'source', '-', 'to', '-', 'source', 'compiler', 'or', 'transcompiler', '.', 'A', 'language', 'rewriter', 'is', 'usually', 'a', 'program', 'that', 'translates', 'the', 'form', 'of', 'expressions', 'without', 'a', 'change', 'of', 'language', '.', 'The', 'term', 'compiler', '-', 'compiler', 'refers', 'to', 'tools', 'used', 'to', 'create', 'parsers', 'that', 'perform', 'syntax', 'analysis', '.'] | {'1.In': 1, 'computing': 1, 'compiler': 9, 'computer': 3, 'program': 5, 'translates': 4, 'code': 4, 'written': 2, 'programming': 2, 'language': 11, 'source': 4, 'target': 1, 'primarily': 1, 'programs': 1, 'translate': 1, 'high': 2, 'level': 5, 'lower': 1, 'e.g.': 1, 'assembly': 1, 'object': 1, 'machine': 1, 'create': 2, 'executable': 1, 'program.[1][2]:p1There': 1, 'different': 2, 'types': 1, 'compilers': 1, 'compiled': 1, 'run': 1, 'CPU': 1, 'operating': 1, 'system': 1, 'runs': 1, 'cross': 1, 'bootstrap': 1, 'intends': 1, 'compile': 1, 'low': 1, 'higher': 1, 'decompiler': 1, 'languages': 1, 'usually': 2, 'called': 1, 'transcompiler': 1, 'rewriter': 1, 'form': 1, 'expressions': 1, 'change': 1, 'term': 1, 'refers': 1, 'tools': 1, 'parsers': 1, 'perform': 1, 'syntax': 1, 'analysis': 1} | {1.In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language).: 7.454545454545454, The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program.[1][2]:p1There: 7.545454545454544, are many different types of compilers.: 0.36363636363636365, If the compiled program can run on a computer whose CPU or operating system is different from the one on which the compiler runs, the compiler is a cross-compiler.: 3.909090909090909, A bootstrap compiler is written in the language that it intends to compile.: 2.2727272727272725, A program that translates from a low-level language to a higher level one is a decompiler.: 2.9999999999999996, A program that translates between high-level languages is usually called a source-to-source compiler or transcompiler.: 3.4545454545454546, A language rewriter is usually a program that translates the form of expressions without a change of language.: 3.3636363636363633, The term compiler-compiler refers to tools used to create parsers that perform syntax analysis.: 2.454545454545454} | The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program.[1][2]:p1There 1.In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language). |
| 3 | Meditation is a practice where an individual uses a technique – such as mindfulness, or focusing the mind on a particular object, thought, or activity – to train attention and awareness, and achieve a mentally clear and emotionally calm and stable state. Scholars have found meditation elusive to define, as practices vary both between traditions and within them.  Meditation is practiced in numerous religious traditions. The earliest records of meditation (dhyana) are found in the Vedas, and meditation exerts a salient role in the contemplative repertoire of Hinduism and Buddhism.[7] Since the 19th century, Asian meditative techniques have spread to other cultures where they have also found application in non-spiritual contexts, such as business and health | ['Meditation', 'is', 'a', 'practice', 'where', 'an', 'individual', 'uses', 'a', 'technique', '–', 'such', 'as', 'mindfulness', ',', 'or', 'focusing', 'the', 'mind', 'on', 'a', 'particular', 'object', ',', 'thought', ',', 'or', 'activity', '–', 'to', 'train', 'attention', 'and', 'awareness', ',', 'and', 'achieve', 'a', 'mentally', 'clear', 'and', 'emotionally', 'calm', 'and', 'stable', 'state', '.', 'Scholars', 'have', 'found', 'meditation', 'elusive', 'to', 'define', ',', 'as', 'practices', 'vary', 'both', 'between', 'traditions', 'and', 'within', 'them', '.', '\n\n', 'Meditation', 'is', 'practiced', 'in', 'numerous', 'religious', 'traditions', '.', 'The', 'earliest', 'records', 'of', 'meditation', '(', 'dhyana', ')', 'are', 'found', 'in', 'the', 'Vedas', ',', 'and', 'meditation', 'exerts', 'a', 'salient', 'role', 'in', 'the', 'contemplative', 'repertoire', 'of', 'Hinduism', 'and', 'Buddhism.[7', ']', 'Since', 'the', '19th', 'century', ',', 'Asian', 'meditative', 'techniques', 'have', 'spread', 'to', 'other', 'cultures', 'where', 'they', 'have', 'also', 'found', 'application', 'in', 'non', '-', 'spiritual', 'contexts', ',', 'such', 'as', 'business', 'and', 'health'] | {'Meditation': 2, 'practice': 1, 'individual': 1, 'uses': 1, 'technique': 1, '–': 2, 'mindfulness': 1, 'focusing': 1, 'mind': 1, 'particular': 1, 'object': 1, 'thought': 1, 'activity': 1, 'train': 1, 'attention': 1, 'awareness': 1, 'achieve': 1, 'mentally': 1, 'clear': 1, 'emotionally': 1, 'calm': 1, 'stable': 1, 'state': 1, 'Scholars': 1, 'found': 3, 'meditation': 3, 'elusive': 1, 'define': 1, 'practices': 1, 'vary': 1, 'traditions': 2, '\n\n': 1, 'practiced': 1, 'numerous': 1, 'religious': 1, 'earliest': 1, 'records': 1, 'dhyana': 1, 'Vedas': 1, 'exerts': 1, 'salient': 1, 'role': 1, 'contemplative': 1, 'repertoire': 1, 'Hinduism': 1, 'Buddhism.[7': 1, '19th': 1, 'century': 1, 'Asian': 1, 'meditative': 1, 'techniques': 1, 'spread': 1, 'cultures': 1, 'application': 1, 'non': 1, 'spiritual': 1, 'contexts': 1, 'business': 1, 'health': 1} | {Meditation is a practice where an individual uses a technique – such as mindfulness, or focusing the mind on a particular object, thought, or activity – to train attention and awareness, and achieve a mentally clear and emotionally calm and stable state.: 9.333333333333332, Scholars have found meditation elusive to define, as practices vary both between traditions and within them.  : 4.333333333333334, Meditation is practiced in numerous religious traditions.: 2.6666666666666665, The earliest records of meditation (dhyana) are found in the Vedas, and meditation exerts a salient role in the contemplative repertoire of Hinduism and Buddhism.[7]: 5.666666666666665, Since the 19th century, Asian meditative techniques have spread to other cultures where they have also found application in non-spiritual contexts, such as business and health: 4.999999999999999 | Meditation is a practice where an individual uses a technique – such as mindfulness, or focusing the mind on a particular object, thought, or activity – to train attention and awareness, and achieve a mentally clear and emotionally calm and stable state. |
| 4 | A stock market, equity market or share market is the aggregation of buyers and sellers of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately, such as shares of private companies which are sold to investors through equity crowdfunding platforms. Investment in the stock market is most often done via stockbrokerages and electronic trading platforms. Investment is usually made with an investment strategy in mind.  Stocks can be categorized by the country where the company is domiciled. For example, Nestlé and Novartis are domiciled in Switzerland and traded on the SIX Swiss Exchange, so they may be considered as part of the Swiss stock market, although the stocks may also be traded on exchanges in other countries, for example, as American depositary receipts (ADRs) on U.S. stock market | ['A', 'stock', 'market', ',', 'equity', 'market', 'or', 'share', 'market', 'is', 'the', 'aggregation', 'of', 'buyers', 'and', 'sellers', 'of', 'stocks', '(', 'also', 'called', 'shares', ')', ',', 'which', 'represent', 'ownership', 'claims', 'on', 'businesses', ';', 'these', 'may', 'include', 'securities', 'listed', 'on', 'a', 'public', 'stock', 'exchange', ',', 'as', 'well', 'as', 'stock', 'that', 'is', 'only', 'traded', 'privately', ',', 'such', 'as', 'shares', 'of', 'private', 'companies', 'which', 'are', 'sold', 'to', 'investors', 'through', 'equity', 'crowdfunding', 'platforms', '.', 'Investment', 'in', 'the', 'stock', 'market', 'is', 'most', 'often', 'done', 'via', 'stockbrokerages', 'and', 'electronic', 'trading', 'platforms', '.', 'Investment', 'is', 'usually', 'made', 'with', 'an', 'investment', 'strategy', 'in', 'mind', '.', '\n\n', 'Stocks', 'can', 'be', 'categorized', 'by', 'the', 'country', 'where', 'the', 'company', 'is', 'domiciled', '.', 'For', 'example', ',', 'Nestlé', 'and', 'Novartis', 'are', 'domiciled', 'in', 'Switzerland', 'and', 'traded', 'on', 'the', 'SIX', 'Swiss', 'Exchange', ',', 'so', 'they', 'may', 'be', 'considered', 'as', 'part', 'of', 'the', 'Swiss', 'stock', 'market', ',', 'although', 'the', 'stocks', 'may', 'also', 'be', 'traded', 'on', 'exchanges', 'in', 'other', 'countries', ',', 'for', 'example', ',', 'as', 'American', 'depositary', 'receipts', '(', 'ADRs', ')', 'on', 'U.S.', 'stock', 'market', '\n'] | {'stock': 6, 'market': 6, 'equity': 2, 'share': 1, 'aggregation': 1, 'buyers': 1, 'sellers': 1, 'stocks': 2, 'called': 1, 'shares': 2, 'represent': 1, 'ownership': 1, 'claims': 1, 'businesses': 1, 'include': 1, 'securities': 1, 'listed': 1, 'public': 1, 'exchange': 1, 'traded': 3, 'privately': 1, 'private': 1, 'companies': 1, 'sold': 1, 'investors': 1, 'crowdfunding': 1, 'platforms': 2, 'Investment': 2, 'stockbrokerages': 1, 'electronic': 1, 'trading': 1, 'usually': 1, 'investment': 1, 'strategy': 1, 'mind': 1, '\n\n': 1, 'Stocks': 1, 'categorized': 1, 'country': 1, 'company': 1, 'domiciled': 2, 'example': 2, 'Nestlé': 1, 'Novartis': 1, 'Switzerland': 1, 'Swiss': 2, 'Exchange': 1, 'considered': 1, 'exchanges': 1, 'countries': 1, 'American': 1, 'depositary': 1, 'receipts': 1, 'ADRs': 1, 'U.S.': 1} | {A stock market, equity market or share market is the aggregation of buyers and sellers of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately, such as shares of private companies which are sold to investors through equity crowdfunding platforms.: 11.833333333333334, Investment in the stock market is most often done via stockbrokerages and electronic trading platforms.: 3.0, Investment is usually made with an investment strategy in mind.  : 0.9999999999999999, Stocks can be categorized by the country where the company is domiciled.: 1.1666666666666665, For example, Nestlé and Novartis are domiciled in Switzerland and traded on the SIX Swiss Exchange, so they may be considered as part of the Swiss stock market, although the stocks may also be traded on exchanges in other countries, for example, as American depositary receipts (ADRs) on U.S. stock market  : 7.333333333333335} | A stock market, equity market or share market is the aggregation of buyers and sellers of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately, such as shares of private companies which are sold to investors through equity crowdfunding platforms. |
| 5 | Engineering is the use of scientific principles to design and build machines, structures, and other items, including bridges, tunnels, roads, vehicles, and buildings.[1] The discipline of engineering encompasses a broad range of more specialized fields of engineering, each with a more specific emphasis on particular areas of applied mathematics, applied science, and types of application. See glossary of engineering.  The term engineering is derived from the Latin ingenium, meaning "cleverness" and ingeniare, meaning "to contrive, devise | ['Engineering', 'is', 'the', 'use', 'of', 'scientific', 'principles', 'to', 'design', 'and', 'build', 'machines', ',', 'structures', ',', 'and', 'other', 'items', ',', 'including', 'bridges', ',', 'tunnels', ',', 'roads', ',', 'vehicles', ',', 'and', 'buildings.[1', ']', 'The', 'discipline', 'of', 'engineering', 'encompasses', 'a', 'broad', 'range', 'of', 'more', 'specialized', 'fields', 'of', 'engineering', ',', 'each', 'with', 'a', 'more', 'specific', 'emphasis', 'on', 'particular', 'areas', 'of', 'applied', 'mathematics', ',', 'applied', 'science', ',', 'and', 'types', 'of', 'application', '.', 'See', 'glossary', 'of', 'engineering', '.', '\n', 'The', 'term', 'engineering', 'is', 'derived', 'from', 'the', 'Latin', 'ingenium', ',', 'meaning', '"', 'cleverness', '"', 'and', 'ingeniare', ',', 'meaning', '"', 'to', 'contrive', ',', 'devise'] | {'Engineering': 1, 'use': 1, 'scientific': 1, 'principles': 1, 'design': 1, 'build': 1, 'machines': 1, 'structures': 1, 'items': 1, 'including': 1, 'bridges': 1, 'tunnels': 1, 'roads': 1, 'vehicles': 1, 'buildings.[1': 1, 'discipline': 1, 'engineering': 4, 'encompasses': 1, 'broad': 1, 'range': 1, 'specialized': 1, 'fields': 1, 'specific': 1, 'emphasis': 1, 'particular': 1, 'areas': 1, 'applied': 2, 'mathematics': 1, 'science': 1, 'types': 1, 'application': 1, 'glossary': 1, 'term': 1, 'derived': 1, 'Latin': 1, 'ingenium': 1, 'meaning': 2, 'cleverness': 1, 'ingeniare': 1, 'contrive': 1, 'devise': 1} | {Engineering is the use of scientific principles to design and build machines, structures, and other items, including bridges, tunnels, roads, vehicles, and buildings.[1]: 4.5, The discipline of engineering encompasses a broad range of more specialized fields of engineering, each with a more specific emphasis on particular areas of applied mathematics, applied science, and types of application.: 6.5, See glossary of engineering.  : 1.25, The term engineering is derived from the Latin ingenium, meaning "cleverness" and ingeniare, meaning "to contrive, devise: 3.75} | The discipline of engineering encompasses a broad range of more specialized fields of engineering, each with a more specific emphasis on particular areas of applied mathematics, applied science, and types of application. |

**Conclusion:**

Automatic summarization is an old challenge but the current research direction diverts towards emerging trends in biomedicine, product review, education domains, emails and blogs. This is due to the fact that there is information overload in these areas, especially on the World Wide Web.Automated summarization is an important area in NLP (Natural Language Processing) research. It consists of automatically creating a summary of one or more texts. The purpose of extractive document summarization is to automatically select a number of indicative sentences, passages, or paragraphs from the original document . Summarization approaches based on Neural Network, Graph Theoretic, Fuzzy and Cluster have, to an extent, succeeded in making an effective summary of a document.Both extractive and abstractive methods have been researched. Most summarization techniques are based on extractive methods. Abstractive method is similar to summaries made by humans.Abstractive summarization as of now requires heavy machinery for language generation and is difficult to replicate into the domain specific areas.

**References:**

[1]Alfonseca, E., Rodr´ıguez, P. Description of the UAM system for generating very short summaries at DUC-2003. In HLT/NAACL Workshop on Text Summarization / DUC 2003, 2003.

[2] Angheluta, R., Moens, M, F., De Busser, R. K.U.Leuven Summarization System. In DUC 2003, Edmonton, al-berta, Canada, 2003.

[3] Aonet, C., Okurowskit, M. E., Gorlinskyt, J., et al. A Scalable Summarization System Using Robust NLP. In Proceedings of the ACL’07/EACL’97 Workshop on Intelligent Sclalable Text Summarization, pages 66-73, 1997.

[4] Barzilay, R., Elhadad, M. Using Lexixal Chains for Text Summarization.In Inderjeet Mani and Mark Marbury,editors, Advances in Automatic Text Summarization. MIT Press, 1999.

[5] Barzilay, R, McKeown, K, Elhadad, M.Information fusionn in the context of multi-document summarization. In Proceedings of ACL 1999, 1999.

1. Boguarev, B., Kennedy, C. Salience based content characterisation of text documents. In Proceedings of ACL’97 Workshop on Intelligent, Scalable Text Summarization, pages 2-9, Madrid,Spain, 1997.

- S. Bharath

2018103519

B.E(CSE)

CEG