UPPSALA
UNIVERSITET

IT 22 060
Degree project 30 credits
June 2022

Information Extraction and Document Similarity: Bag-of-Concepts based approach

Shubhomoy Biswas

Master's Programme in Computer Science

UPPSALA
UNIVERSITET

Information Extraction and Document Similarity: Bag-of-Concepts based approach

Shubhomoy Biswas

Abstract

People in many organizations develop rich-text files, such as Microsoft Word (MS-Word) and Microsoft Powerpoint (MS-Powerpoint), which contain textual content in a variety of domains, from product presentations to confidential paperwork. This thesis examines information extraction methods, provides a concept-based strategy for computationally representing documents, and determines the degree of similarity between documents based on the information contained in them. Finally, the proposed method of document representation's future scope is examined, as well as how it might be applied to various text/data mining approaches. The thesis is completed in an organization (Ericsson AB) where the proposed approach is tested on a genuine set of documents.

Faculty of Science and Technology

Uppsala University, Uppsala

Supervisor: Roger Persson Subject reader: Georgios Fakas

Examiner: Mats Daniels

Acknowledgements

I would like to thank my primary supervisor - Roger Persson and my manager Alexandra Darulova at Ericsson AB who provided me all the necessary information and tools that I needed to carry out my thesis work smoothly in the organisation. They allowed me to setup regular meetups, provide contacts, and guided me throughout my studies. I would also like to thank my reviewer - Georgios Fakas at Uppsala University who reviewed my work and my report at major milestones. He provided me with insights and structure so that I can plan my studies accordingly. Lastly, to all my friends who supported me in evaluating my work and peer reviewing at the University.

Contents

1 Introduction

1.1 Setting and Problem Statement

The thesis is conducted in an organisation - Ericsson AB. Ericsson AB is a multinational networking and telecommunication company having it's headquarter in Stockholm, Sweden. At it's core, the organisation offers infrastructure, software and services in Information and Communication Technology for telecommunication service providers. Multiple teams inside Ericsson AB are involved in various projects ranging from internal research, studies, and development to customer-centric solutions and services.

Having a varied set of work areas and services, digital files such as rich-text documents are constantly being created by the people at Ericsson AB on a daily basis. For future reference, these documents are often uploaded and saved in a shared on-premise cloud repository. This cloud-based file repository makes it simple to save, search, and share historical information across employees. However, there are no restrictions on the writing style, template, or type of files that can be uploaded. Due to the fact that several teams are working on various projects at the same time, there are sometimes overlaps in domain areas, ideas, and resources noted in these files. They are frequently unknown to the teams provided a common repository with a plethora of files and directories. In this situation, the difficulty of determining what else exists in a given domain area becomes a hurdle.

Furthermore, any cloud file repository, in general, is frequently updated by the authors of the files in question, and thus is prone to become complex as a result of unstructured file directories, diverse data, and duplicates. In a real-world scenario, for example, when it comes to referencing these files, one must have the domain knowledge of the document in question as well as a working knowledge of the repository structure in order to locate the appropriate information. In other circumstances, retrieving documents that are comparable to the sought information is a constant issue.

1.2 Purpose and Scope

The purpose of this thesis is to,

1. Analyse few, if not all, Ericsson AB's internal documents and figure out a solution to better retrieve information from these files.
2. Propose a mechanism for determining the connections between the documents in this repository, and
3. Find out how similar any two documents are in terms of the content they contain if a relationship exists between them.

To satisfy the above, the following scope is defined,

1. Given a query document from a set of documents[1] as input, propose other documents from the same set that are similar or related based on the information they contain.
2. In order to perform #1, first study various document representation methods.
3. Explain how concept-based approach of document representation fits with the current use-case.

1.3 Report Structure

To address the problem at hand in a real world scenario, various studies have been conducted, a possible method of document representation is proposed along with the implementation to satisfy the purpose of the thesis. In Section 2, the current setting and challenges are discussed. Section 3 discusses a few related methods of document representation followed by Section 4 which explains the proposed method in this thesis. Samples of data that is used is stated and explained in Section 5. Section 6 tries to explain with example how a document can be represented with the proposed method. Experiments with their results are discussed in Section 7, and finally Section 8 contains possible future extensions.

2 Background

In text and/or data mining methodologies, information retrieval (IR) is a significant area [7]. IR systems can be used for a variety of tasks, including document similarity detection, responding to user queries based on document content, text summarization, and more. From the problem statement discussed in 1.1, because rich-text files are often written as free text with varied writing styles, an IR system is the first and foremost necessity for solving this use-case. Key information can be acquired using the IR approach in order to either understand the document's domain and establish relationships with other documents (similarity) or to retrieve responses and information for a particular query from these documents [1].

Any information retrieval approach, however, is dependent on the initial step, namely, how documents are represented computationally. This is extremely important because it determines the method's overall effectiveness and performance. The document representations are also critical in determining how we wish to approach specific use-cases when dealing with various files and the data they contain in general.

[1] The set of documents will refer to Ericsson AB's file repository

For example, below are three files (MS-Word type) containing study proposal from from the set[1]. Each one addresses specific domain area. Due to privacy concerns, a gist of the scope is mentioned and few terms are obscured.

1. **Study Proposal Log Access.docx** - *"Propose a standardized way how designers and testers can effectively access logs, from Production System according to the security regulations. The study shall identify which logs and other related information that need to be accessible for designers and testers."*
2. **Study Proposal PS Monitoring Strategy.docx** - *"Create a Monitoring Strategy for monitoring the health of the Production System. The Strategy targets the developers of the production system, to be able to provide appropriate monitoring tools for their respective components. The study relates to monitoring production system health, through collecting, aggregating and visualizing data from components and tracking and measuring the flow of request execution through multiple applications and interfaces."*
3. **Study Proposal *** CI Storage.docx** - *"Current CI loops does not have a formalized and structured way of storing logs for analytics purposes. *** would like store CI data from their loops to enable ML and drive data driven approaches. Currently this data is not being stored in a structured way."*

Employees frequently write documents like these as free text, using their own writing style, template, grammar, and language. However, each of them focuses on a different technical aspect of Ericsson AB. They could be connected in terms of subject expertise, but otherwise be utterly unrelated. Documents 1 and 3 in the preceding example are more aligned with application log management, which includes extracting, storing, and handling system data logs, whereas Document 2 is primarily concerned with monitoring systems. Documents 1 and 3 with regard to log management might be grouped together as a possible solution to our problem, while document 2 could have been similar in other documents in the set with monitoring-related subjects.

2.1 Challenges

A file repository with various textual files, like the example above, can hold thousands of documents, depending on the size of the company. There are numerous departments in Ericsson AB, each of which is focused on a different aspect of telecommunications and IT, and each department has multiple teams. They use an internal file repository to store these documents, plans, presentations, and other information that is useful in their daily work or for future reference. This posses few challenges,

1. Hard to identify overlaps of ideas and documentations addressing specific areas when many teams are working in parallel on different projects.

2. It becomes a complicated task to find out what else exists that are related to a specific domain.
3. When the size of the repository having these information is large, it adds to the complexity of #1 and #2.
4. There are also possibilities of having duplicate files or similar files (eg: same file with different update versions) that can add redundancy and make the task of searching for information more complicated.

2.2 Research Question

Given the background and the challenges, this paper proposes what methods of document representation and information retrieval systems should work in the current setting. specifically,

- How can these rich-text documents be computationally represented?
- Given a document from a set of documents, how can the *proposed* document representation method assist in retrieving other related documents, w.r.t the content, domain and/or topic, from the set?

3 Related Work

Documents must be represented in such a way that a system can comprehend the linguistic features of the textual data and hence discriminate between them based on their content. The techniques and algorithms used for performing modelling of these textual data requires a fixed length inputs and outputs. Therefore, vectors are derived from textual data in order to reflect various linguistic properties [3]. In the current use-case, understanding these linguistic qualities is useful since diverse writing styles are used in these documents that address distinct parts of organizational initiatives and job areas. The following are some related methods for representing documents computationally.

3.1 Bag-of-Words (BoW)

The Bag-of-Words describes a document by counting the number of times certain words appear in it. The order of the defined terms in the documents is ignored, leading to the term "bag" of words. The logic is straightforward: any two documents are considered comparable if they include roughly the same number of vocabulary words from a defined collection. The difficulty lies in deciding which vocabulary terms to use to represent a large document repository and scoring the presence of each word in those documents.

3.2 Word2Vec

Word2Vec embeds words in continuous vector space using a basic neural network. The weights of the neural network are adjusted during training so that the word2vec skip-gram model predicts the surrounding words of a given input word within a specific window size. Words that appear in comparable contexts have similar meanings, according to the distributed hypothesis [5], and their embeded vector representations are close to each other. The generated representation of words can be subjected to a variety of machine learning and data mining techniques. In the bag-of-word technique, the sparsity and dimension can skyrocket, but word2vec produces a reasonable dimension for building dense document vectors. A simple approach for representing a document is by averaging the word vectors that are in the document [8].

3.3 Doc2Vec

Doc2Vec natively embeds documents in a continuous vector space. It uses a neural network in the same way as word2vec does, but it also includes documents in the input layer. In comparison to the averaging method of word2vec [2], the representation power of doc2vec has demonstrated to be very effective in document categorization and clustering. However, in order to grasp the reasoning behind the output of any data mining approach, it is necessary to have an intuitive comprehension of the document representation. Doc2vec is unable to do so.

3.4 Concept based approach

When it comes to document representation and feature extraction, the BoW (Bag-of-Words) technique has two fundamental flaws. In the current scenario, where the number of documents is likely to increase, the defined set of terms expands, increasing the dimension by which the documents are represented. Additionally, sparcity in the representation is introduced, which can have an impact on the underlying processes, such as data mining or machine learning activities. Second, despite the fact that words in documents can have similar meanings and contexts, BoW recognizes all words as independent. On the other hand, as discussed before, doc2vec fails to provide an intuitive understanding of the resultant document vectors even though it showed promising results in classification and clustering tasks. Bag-of-Concepts (BoC) get around these constraints by clustering Word2Vec's dispersed representation of words [6]. This method groups semantically similar words into common concepts, which are subsequently represented in documents by the frequency with which they contain the concepts. This limits the dimension of expressing the documents to the number of concepts while keeping semantically comparable phrases together. The weights of these concepts are used to create the resultant document vectors, which show how each document links to the concepts and to what extent.

4 Proposed method for the use case

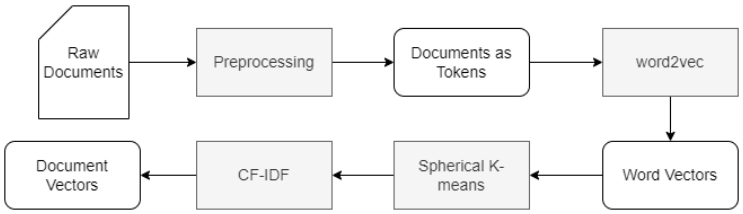


Figure 1: Flow of Concept-based method of document representation.

The concept-based strategy is proposed in this study for embedding documents into M-dimension vector space, where M specifies the number of concepts. Here, concepts denotes the clusters of word vectors given by the word2vec approach. First the textual elements from raw documents of type MS-Word and MS-Powerpoint are preprocessed into individual tokens and common english stopwords are removed. A word2vec model is trained on these tokens to generate the vector representation of each token in N-dimension space. Following the vector representation of the words, Spherical K-means is used to group them into clusters. As a result, these groups are referred to as "Concepts." These Concepts are then known to hold tokens that are contextually similar to one another. Like the Bag-of-Words method, the documents will be represented as vectors of these Concepts where each dimension measures the frequency of the corresponding Concept they are holding. After that, the documents are weighted using Concept Frequency-Inverse Document Frequency (CF-IDF), and vector representations of them are created.

$$CF - IDF(c_i, d_j, D) = \frac{n_c}{\sum_k n_k} log \frac{|D|}{|\{d \in D | c_i \in d\}|} \tag{1}$$

Here, $|D|$ is the number of documents in the set, the denominator is the number of documents from the collection `D`, in which the concept c occurs; n_c is the number of occurrences of concept c in document d, and n_k is the total number of concepts in this document.

Features of this type of document embedding becomes intuitive as each dimension, or the degree by which they are linked to each concept, contains semantically similar words.

5 Data Description

The data used here are files of type MS-Word and MS-Powerpoint uploaded in Ericsson AB's private repository. Not all files of these types are considered due to accessibility rights and data sensitivity. We limit our data requirements for this thesis to Ericsson AB's *Study Proposal* and *Technical Report* documents with few other *Guidelines* and *Principles* documents.

6 Explanation with Examples

6.1 Documents Preprocessing

All documents in the set are preprocessed before applying any downstream method. The desired outcome after the preprocessing are a list of tokens for each document.

An example line from an architectural design document inside Ericsson AB's file repository,

```
This document identifies the artifacts needed to describe an Architecture
   Pattern in The Ericsson Architecture
```

is finally converted to a list of lowercase tokens excluding the stop words.

```
['document', 'identifies', 'artifacts', 'needed', 'describe',
   'architecture', 'pattern', 'ericsson', 'architecture']
```

The preprocessing stages include:

1. Extraction of textual data from the rich-text documents, that is MS-Word and MS-Powerpoint files.
2. The texts are converted to lowercase to maintain consistency.
3. English punctuation are removed.
4. Removal of English stop words.
5. Word tokenization

6.2 Word Embedding

A `word2vec` neural network model is trained using all the tokens from the set of documents as text corpus. In the higher dimension vector space, a `word2vec` neural network model generates a vector representation for each token. This is referred to as `word embedding`. Hyperparameters are often defined as dimensions in the range of 100 to 300.

For example, after training the model using vector size of 10 (dimensions), the word embeddings of "RAN" (acronym of Radio Access Network) and "Study" are as follows. Note that a dimension of 10 is chosen as an example.

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
RAN	0.413	0.110	1.204	-0.647	-0.321	0.542	-0.252	-0.230	-0.010	-1.171
Study	-0.013	0.612	0.892	-0.893	0.447	0.874	-0.085	-0.328	0.379	0.041

Table 1: Vector representation in 10 dimension vector space for the tokens - "RAN" and "Study" after training a `word2vec` model.

Word embedding is performed in order to learn the contextual similarity between each pair of tokens. To put it another way, if two tokens are used together in the same context (or near each other), their word vectors should be similar. The *cosine similarity* angle is a numerical representation of the same.

The top 10 comparable tokens of "RAN" (Radio Access Network) and "Study" respectively are presented below with their *cosine similarity* measure after training a `word2vec` model with a vector size of 300 (output dimensions).

Token	Cosine Similarity	Token	Cosine Similarity
cloud	0.883191	msrbs	0.806189
plane	0.835507	positioning	0.794724
smo	0.830831	conformance	0.794388
exilis	0.815087	oran	0.793737
orusim	0.808042	rbs	0.789409

Table 2: Top 10 tokens similar to "RAN" (Radio Access Network)

Token	Cosine Similarity	Token	Cosine Similarity
studies	0.820978	proposal	0.764747
inputs	0.786150	executive	0.752705
participants	0.770291	anatomy	0.743834
ioa	0.766440	psab	0.742869
scope	0.765401	ngp	0.740924

Table 3: Top 10 tokens similar to "Study"

6.3 Bag of Concepts

The corresponding word embeddings are then grouped together as clusters called "Concepts". Spherical K-Means was used and tried out with different values of K. This method provides a solution to reduce the large dimensions often generated by traditional method like TF-IDF or Bag-of-Words.

In the above example, "`cloud`" and "`plane`" ended up in the same cluster.

For example, with a word embedding of dimension 300 and then creating 10 concepts by spherical k-means clustering, the CF-IDF of any document can be represented by a vector of 10 dimensions. As an example, a document named - `Study Proposal Product Improvement Database.docx` from the set [1] is represented as follows

```
Study Proposal Product Improvement Database.docx
[3.12 1.61 3.86 0.47 1.11 2.64 1.08 1.29 6.18 0.59]
```

This example shows us that how the higher dimension of 300 used for word embedding is reduced to 10 in order to represent the document vector. Each of the dimension measures the frequency by which the concept is being discussed in the document. And we already know that concepts are a cluster of semantically similar words grouped together.

7 Experiment Results

Here we discuss the results of the proposed Bag-of-Concept based method experimented over real documents at Ericsson AB. The section also provides how the representation of these documents can be interpreted in order to better and intuitively understand the logic behind this interpretation.

7.1 Interpretability

The Bag-of-Concept approach provides an alternate way to interpret the representation of the documents. To understand these representations, we embedded the words into a continuous space of 300 dimensions using `word2vec` model and then grouped them into 10 concepts. Each document vector is represented by these 10 concepts. The individual dimension of the vectors signifies the weight given to the corresponding concept. It then becomes easier to visualize how much the documents are related to the contextual relationship with the words that define these concepts.

We take two documents as an example from the set [1] containing similar domain-specific content.

1. Document 1 defines a "Study Proposal Product Improvement Database".
2. Document 2 defines a "Study proposal Monitoring strategy"

Each of their vector representation with respect to 10 concepts are shown below.

	Concept 0	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5	Concept 6	Concept 7	Concept 8	Concept 9
Document 1	3.12	1.61	3.86	0.47	1.11	2.64	1.08	1.29	6.18	0.59
Document 2	2.78	0.98	3.57	0.68	0.51	2.15	0.58	1.17	7.42	0.83

Table 4: Document vectors represented in 10 dimensions or 10 Concepts

We can see from the table above that both of these documents have higher and very equivalent weights in the Concept 2. Below are the top 20 word embeddings that are closest to the centroid of Concept 2. The distance from the centroid is measured using cosine similarity.

Word	Cosine Distance	Word	Cosine Distance
close	0.98	break	0.97
effective	0.97	bring	0.97
spent	0.96	years	0.96
turn	0.96	self	0.95
automate	0.95	demand	0.95
especially	0.95	months	0.95
consequences	0.94	depends	0.94
agree	0.94	developing	0.94
involve	0.94	achieve	0.94
fte	0.94	achieved	0.94

Table 5: Top 20 words whose corresponding vectors are closest to Concept 2 centroid.

Both documents appear to be about time spans (keywords: `years`, `months`), development (keywords: `developing`, `automate`), efficacy, and efficiency (keywords: `achieve`, `effective`), based on the equivalent phrases in Concept 2. Words like "automate," "effective," and "accomplish" appear to indicate that these texts are about improvement and delivery plans. We can, on the other hand, look at concepts where Document 1 and Document 2 have contrasting weights. Document 1 has a higher weight in Concept 6 than Document 2, while Document 2 has a higher weight in Concept 8 than Document 1. The contents of Concept 6 and Concept 8 are listed below.

Word	Cosine Distance	Word	Cosine Distance
docs	0.96	viewpage	0.96
eteamproject	0.95	forum	0.95
online	0.95	html	0.95
browse	0.95	yang	0.94
eteamspace	0.94	master	0.94
pdl	0.93	transformation	0.93
alliance	0.93	onboarding	0.92
display	0.92	www	0.92
sharepoint	0.92	mathias	0.92
niklas	0.92	wiki	0.91

Table 6: Top 20 words whose corresponding vectors are closest to Concept 6 centroid.

Few words in Concept 6, such as "association," "forum," "transformation," "onboarding," and "wiki," offer us a rough notion of what this concept entails. It appears that the concept is

primarily focused on documentation, member onboarding, discussion forums, and other activities that may be relevant to product enhancement as specified in Document 1 but not in Document 2.

Word	Cosine Distance	Word	Cosine Distance
chapters	0.97	epics	0.95
completed	0.94	think	0.94
identifying	0.93	respective	0.93
detail	0.93	drive	0.93
discussions	0.93	improvement	0.92
testable	0.92	lists	0.92
ontology	0.92	idea	0.92
evolve	0.92	modelling	0.92
aligned	0.92	architectural	0.92
proposals	0.92	clearly	0.92

Table 7: Top 20 words whose corresponding vectors are closest to Concept 8 centroid.

The words "identifying," "testable," "epics," "improvement," "modeling," and "architectural" in Concept 8 refer to internal Ericsson AB conversations and documentation about product strategies and goals. As a result, Document 2 has a higher Concept 8 weight than Document 1.

Apart from comparing similar documents to see how they differ, we can also study individual documents to see what key domain area they focus on. To show this, an internal document defining application log ingestion-"`PS Log Ingestion.docx`" was used. Below is a vector representation of this document in terms of the ten concepts.

Document 3 below corresponds to `PS Log Ingestion.docx.docx`

	Concept 0	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5	Concept 6	Concept 7	Concept 8	Concept 9
Document 3	0.92	1.24	1.01	0.43	6.11	1.27	0.1	0.55	2.44	0.26

Table 8: Vector representation of PS Log Ingestion.docx w.r.t 10 Concepts

We can observe that this document has the most weight in Concept 4. The top 20 words that are most similar to Concept 4 are displayed below.

Word	Cosine Distance	Word	Cosine Distance
retrieve	0.97	ingested	0.97
decryption	0.97	kibana	0.96
duplicated	0.96	scanner	0.95
raw	0.95	storing	0.95
parse	0.95	archiver	0.95
mhweb	0.94	processed	0.94
notification	0.94	dcgm	0.94
parsed	0.94	apis	0.94
archive	0.94	debugger	0.94
visualisation	0.94	download	0.94

Table 9: Top 20 words whose corresponding vectors are closest to Concept 4 centroid.

Many terminology in Concept 4 are connected to application logging jargon, such as "raw," "parse," "archive," "kibana," "storing," "debugger," and so on. This suggests that the document's content is indeed linked to logging architecture and pipelines.

We also observe that the approach is capable of grouping people (usernames) into a single concept (Concept 1 in this case), which is intriguing.

Word	Cosine Distance	Word	Cosine Distance
ann	0.99	charlotte	0.98
orosz	0.98	arvid	0.98
andreas	0.98	karlsson	0.98
olsson	0.98	eriksson	0.98
dennis	0.98	nilsson	0.98
lindgren	0.97	schüller	0.97
lenasson	0.97	tobias	0.97
hallmen	0.97	persson	0.97
ekdahl	0.97	karin	0.97
roger	0.97		

Table 10: Top 20 tokens closest to the centroid of Concept 1

7.2 Real Scenario Evaluation

Our first research question (2.2) is answered by the concept-based document representation. This technique of document representation enables us to better comprehend document vectors and the extent to which they represent each concept.

We applied this strategy to answer our second research question. We computed the cosine similarity between a query document and other documents in the set[1]. Since all the documents are now represented by M-dimensional vectors (where M = number of concepts), a greater cosine similarity score with regard to the query document allows us to retrieve documents from the set that are similar or related to the query document.

For example, we take the same document - `PS Log Ingestion.docx` from the set[1] as a query. The cosine score is then generated by comparing it to other documents in the repository using it's vector representation of 10 concepts. Table (11) and Table (12) display the top five highest candidates, as well as their vector representations. In addition, the five candidates with the lowest scores are listed in Table (13), along with their vector representations in Table (14).

Document	Similar Documents	Cosine Score
Query Document	PS Log Ingestion.docx	
Document 1	PS ESI Dataflow Improvements.docx	0.986624
Document 2	PS Streaming Data Ingestion.docx	0.986566
Document 3	OA Node data ecosystem evolution &...	0.981542
Document 4	NosaLegacyMigrationPss.docx	0.973707
Document 5	NosaLegacyMigrationOa.docx	0.971192

Table 11: Top 5 related documents with highest cosine score with query document.

	Concept 0	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5	Concept 6	Concept 7	Concept 8	Concept 9
Query Document	2.34	1.50	2.27	1.05	6.38	1.71	0.25	0.95	2.85	0.29
Document 1	2.46	1.71	3.94	0.94	13.78	4.19	0.57	1.92	3.86	0.24
Document 2	1.03	0.63	0.80	0.43	7.47	1.36	0.50	0.68	2.04	0.08
Document 3	0.84	0.30	0.45	0.27	4.71	0.60	0.06	0.29	1.61	0.06
Document 4	6.10	3.18	4.01	2.84	29.96	3.89	2.49	2.01	5.63	0.57
Document 5	2.41	0.20	5.18	0.19	0.04	1.32	2.09	0.29	3.21	0.44

Table 12: The vector representation of the documents mentioned in Table (11).

Document	Unrelated Documents	Cosine Score
Query Document	PS Log Ingestion.docx	
Document 1	FEP-313 Study.docx	0.365539
Document 2	Study Proposal RITTS Engineering Environment.docx	0.356074
Document 3	OA for BID-13513 GTMS - TGF - Eiffel support.docx	0.325934
Document 4	ORAN LLS Tools Strategy.docx	0.257698
Document 5	proba123.docx	0.132699

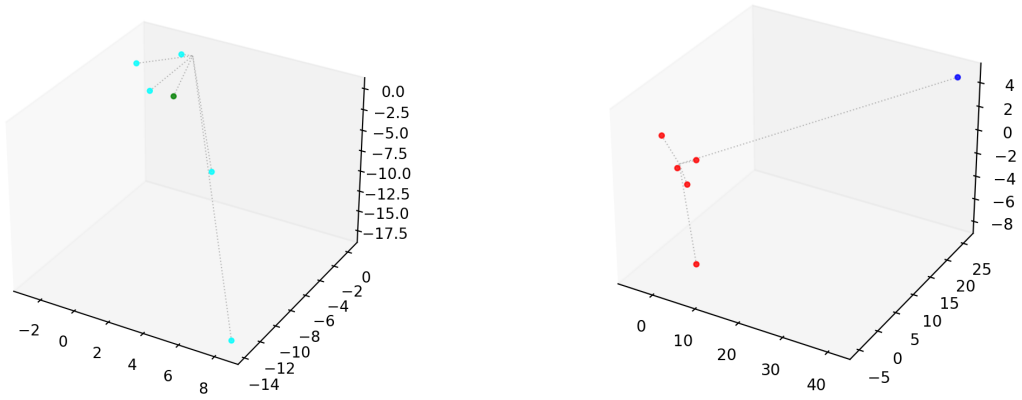
Table 13: 5 documents having the least cosine score with the query document.

	Concept 0	Concept 1	Concept 2	Concept 3	Concept 4	Concept 5	Concept 6	Concept 7	Concept 8	Concept 9
Query Document	2.86	3.70	11.25	4.99	2.05	8.94	1.09	50.19	9.76	3.67
Document 1	0.56	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.01
Document 2	1.12	0.80	1.76	18.22	3.27	3.34	0.24	2.53	2.73	0.54
Document 3	3.81	8.55	5.76	1.23	0.98	4.73	15.61	3.84	6.76	2.34
Document 4	2.32	0.59	5.18	0.77	0.85	3.55	0.58	1.17	6.45	1.17
Document 5	2.41	0.20	5.18	0.19	0.04	1.32	2.09	0.29	3.21	0.44

Table 14: The vector representation of the documents mentioned in Table (13).

We can go further into each dimension (or concept) of the document vectors to see how much they represent each concept, as discussed in Section (7.1). The concept-based document representation can also be viewed in two or three dimensions. Because the document representation in this scenario utilized ten concepts (or dimensions), PCA (Principle Component Analysis) was used to reduce the number of dimensions to three and then visualized in three-dimensional space.

In Fig (2a), the green dot denotes the query document - *PS Log Ingestion.docx* while the light blue dots are the other 5 documents that the method found similar or related. On the contrary, Fig (2b) shows 5 other documents (with red dots) that are having the least cosine similarity score.



(a) Query document with 5 similar documents. (b) Query document with 5 least similar documents.

Figure 2: Visualisation of the vector representation of the documents.

The preceding visual representation (2) shows how the documents tend to be closer in angle to the query document, while the least similar or irrelevant documents try to stretch out.

Each document can be queried in the same way to find other documents in the set that have a strong concept-wise relationship with the query. This allows us to visualize the repository space in greater detail. When the aforesaid technique is used to a subset of documents, the visualisation below is obtained. The greater the cosine similarity score between two documents, the broader the edges connecting them.

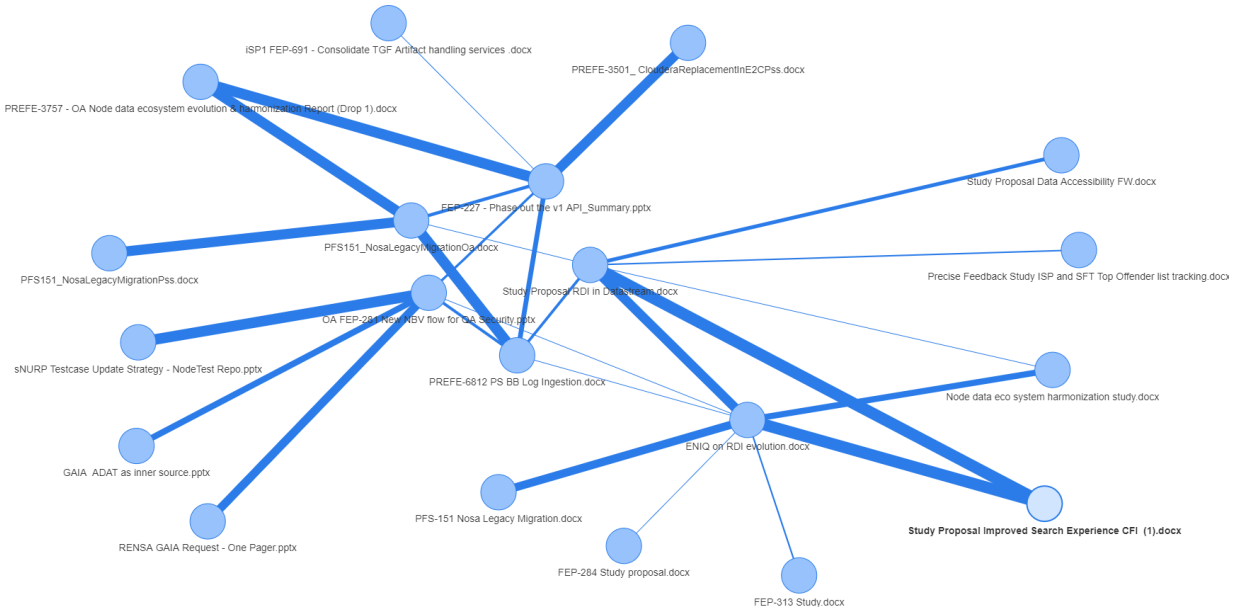


Figure 3: A sample visualisation showing the concept-level relationship between a subset of documents.

Survey Result

A survey was conducted within Ericsson AB to determine the effectiveness of measuring document similarity using the proposed method in a real-world context. Eighteen documents from the set [1] were evenly divided into three groups of six documents in each group for the survey. The top 6 similar documents retrieved by the proposed approach given a query document from the repository are represented by the 6 documents in each group. Employees at Ericsson AB were asked to rate the similarity of each pair of documents on a scale of 1 to 10 (inclusive). They were instructed to do so based on the content and major domain areas indicated in the documents. The documents utilized in each group are listed in Tables ([15]), ([16]), ([17]). The resultant similarity scores are re-scaled from 0-10 to 0-1 and are shown in Figure([4]), ([5a]), and ([5b]).

Doc. No.	Document
1	PS Log Ingestion.docx
2	OA Consolidate PM Ingest Datastream.docx
3	PS ESI Dataflow Improvements.docx
4	NosaLegacyMigrationOa.docx
5	Study Proposal RDI in Datastream.docx
6	Study Proposal Data Accessibility FW.docx

Table 15: Documents present in Group 1

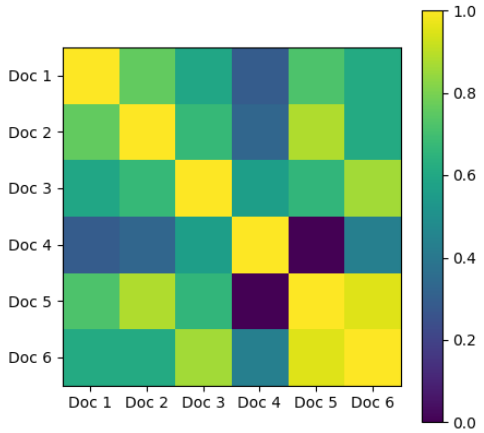
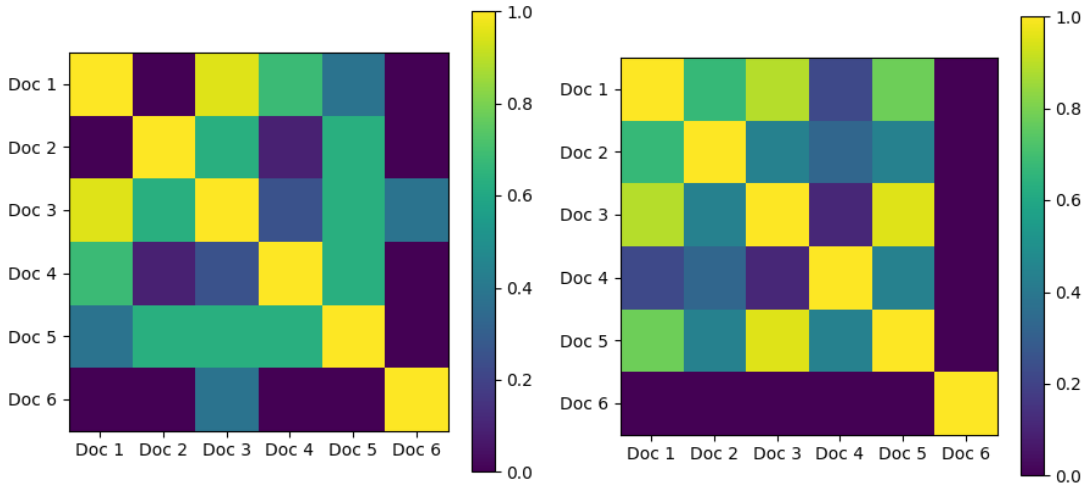


Figure 4: Similarity score matrix between each pair of documents from Group 1.

Doc. No.	Document
1	Study Proposal Product Improvement Database.docx
2	System Principles and Expectations on Product Enabling an Efficient Production.docx
3	Study Proposal Data Accessibility FW.docx
4	Precise Feedback Study ISP and SFT Top Offender list tracking.docx
5	Node data eco system harmonization study.docx
6	FEP-313 Study.docx

Table 16: Documents present in Group 2



(a) Group 2

(b) Group 3

Figure 5: Similarity score matrix between each pair of documents from Group 2 and Group 3.

Doc. No.	Document
1	The Prospect of a Soft FSUE.docx
2	TE Analysis Report MR8891 4490.docx
3	TE Analysis Report for CSIM and LTEsim_UCtool.docx
4	SP_MR38_1099_presentation.pptx
5	ORAN LLS Tools Strategy.docx
6	CIL2978_MCT_Using_RCSSIM_Study_Report.docx

Table 17: Documents present in Group 3

We try to capture what individuals at Ericsson AB think about the relationship between the pair of documents in each set based on the survey results and compare that to the conclusion of the method. Based on these findings, we can conclude that the suggested method aids in the discovery of similarities between documents utilizing the underlying Concept-based representation, as shown in Group 1[4], as well as the ranking of such documents. However, the results in Group 2[5a] and Group 3[5b] demonstrate that people's perspectives and opinions are highly subjective. To put it another way, some people may regard any two documents as being similar in relation to "Domain X," while others may see them as wholly distinct. As a result, it is safe to assume that the proposed approach won't produce a clear and correct outcome in terms of ranking or discovering related documents that meets everyone's needs. However, we discovered that the proposed method aids us in identifying crucial nuances, or domain areas, inside these documents

in order to build a similarity relationship between them, which would otherwise go unnoticed. Also, as we established in Section 7.1, the representation of any document utilizing the said Concept-based approach might assist us to understand how it was written by looking at the final representation and justifying it.

7.3 Effectiveness

Apart from the survey, a classification task similar to Dai et al. [2] was performed. Due to the fact that classification requires labelled dataset to operate upon, we used the BBC-dataset [4] that contains text corpus from various news articles. The documents in this collection are organized into triplets, with two documents belonging to the same type of news story and the third document belonging to a separate category. The cosine similarity between each pair from this triplet is calculated, and the classification is marked true if the score is higher between those two documents that belong to the same category. There are 2,225 text documents in the BBC collection, which are divided into five categories: business, entertainment, politics, sports, and technology. There are 487,434 unique tokens in the entire set after preprocessing, with an average document length of about 220 tokens.

Moreover, there are numerous hyperparameters that can affect the classification task's performance. To assess the classification's efficacy, we opted to modify and compare the following essential parameters, which are the most important in this suggested document representation method:

1. The number of dimensions with which each token will be represented when training the Word2Vec model. In other words, the dimensions for word embedding.

2. The number of concepts with which each document will be represented.

The other hyperparameters where fixed and shared between all the experiments. The windows size was set to 9 and training epochs to 3 in Word2Vec model training. Word2Vec from the Gensim[2] library in python was used for the following experiment. Different embedding dimensions of the Word2Vec model were tested starting from 100 dimensions to 500 dimensions with increments of 100 dimensions. For each of the specified dimension, different number of concepts were tested out in this classification task. The comparison between the tests with their F1-score are listed below.

[2] https://radimrehurek.com/gensim/

	Concept 10	Concept 50	Concept 100	Concept 150	Concept 200	Concept 250	Concept 300	Concept 350	Concept 400
100 Dimensions	0.582	0.568	0.548	0.598	0.000	0.609	0.631	0.592	0.612
200 Dimensions	0.630	0.000	0.000	0.586	0.592	0.599	0.606	0.594	0.617
300 Dimensions	0.650	0.516	0.000	0.572	0.629	0.609	0.608	0.597	0.602
400 Dimensions	0.000	0.000	0.565	0.578	0.601	0.604	0.628	0.614	0.666
500 Dimensions	0.622	0.000	0.552	0.563	0.569	0.597	0.625	0.649	0.619

Table 18: A comparison chart of F1-scores from the classification task w.r.t different word embedding dimension and number of concepts

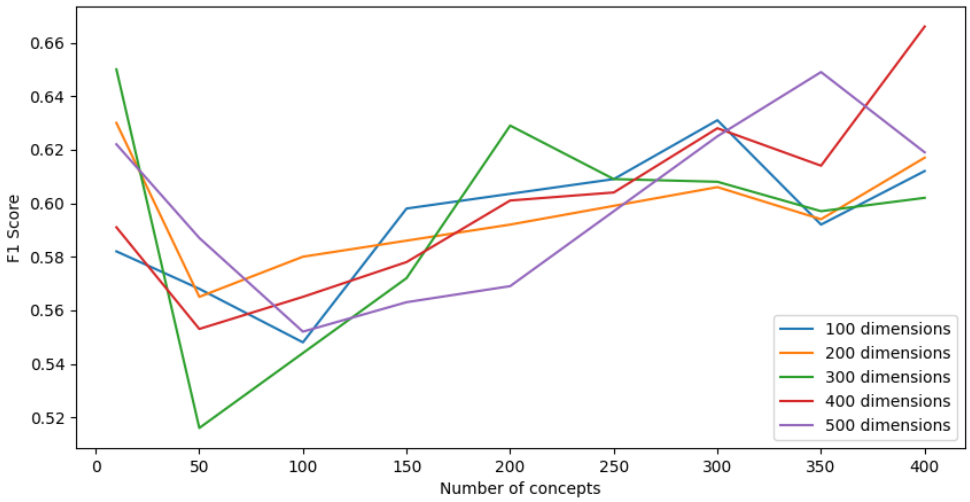


Figure 6: F1-scores of document classification on BBC-dataset.

The evaluation shows two key findings. First, increasing the dimensions of word embedding, used for training the `word2vec` model did not give satisfactory improvements to the F1-score. Rather, the graph for all the tried out dimensions, look similar. Second, as the number of concepts increased, so did the F1-score. This could be because the documents are being represented more efficiently as the number of concepts required to define them grows.

7.4 Time Complexity

There are numerous parameters in play to determine the time complexity of the proposed approach. Mainly there are three key parts where computation time is dependent on,

1. Training the `word2vec` model to retrieve word embeddings for the tokens.
2. Creation of concepts (clusters) using `spherical k-means`.
3. Calculating the CF-IDF for each document in the set.

The word2vec model from gensim library is used to calculate the vector form of each unique tokens from the text corpus. It uses a binary-search over a vocabulary-sized array to achieve the sampling of negative examples, so its time complexity might technically be:

$$O(N * log(V))$$

where N is the size of the corpus and V is the number of unique words in the vocabulary. In practice, the time complexity more generally depends linearly on the size of the corpus.

For spherical k-means, the running time of Lloyd's algorithm and other variants mostly is

$$O(nkdi)$$

where n is the d-dimensional word vectors, k is the number of clusters and i is the number of iterations required. And the time complexity of CF-IDF depends linearly on the size of text corpus.

In order to practically determine how much time it requires to convert all documents in the BBC-dataset to their concept-based embedding, we carried out the same tests as above with increasing number of word embedding dimensions and concepts. The experiement is performed on a 64-bit operating system having Intel i7 2.3GHz processor (x64-based) with 32 GB of RAM.

	Concept 10	Concept 50	Concept 100	Concept 150	Concept 200	Concept 250	Concept 300	Concept 350	Concept 400
100 Dimensions	12.96	12.86	12.23	14.32	14.51	14.69	15.09	15.74	16.60
200 Dimensions	23.80	11.90	18.75	25.60	24.68	25.53	26.83	26.13	27.24
300 Dimensions	33.42	32.61	33.94	35.26	34.78	36.64	38.77	34.64	33.29
400 Dimensions	27.69	34.46	41.24	42.95	45.70	51.25	45.01	54.81	55.37
500 Dimensions	57.44	56.38	55.31	55.34	53.41	61.58	63.49	65.23	70.36

Table 19: Computation time in seconds w.r.t number of embedding word dimensions and concepts.

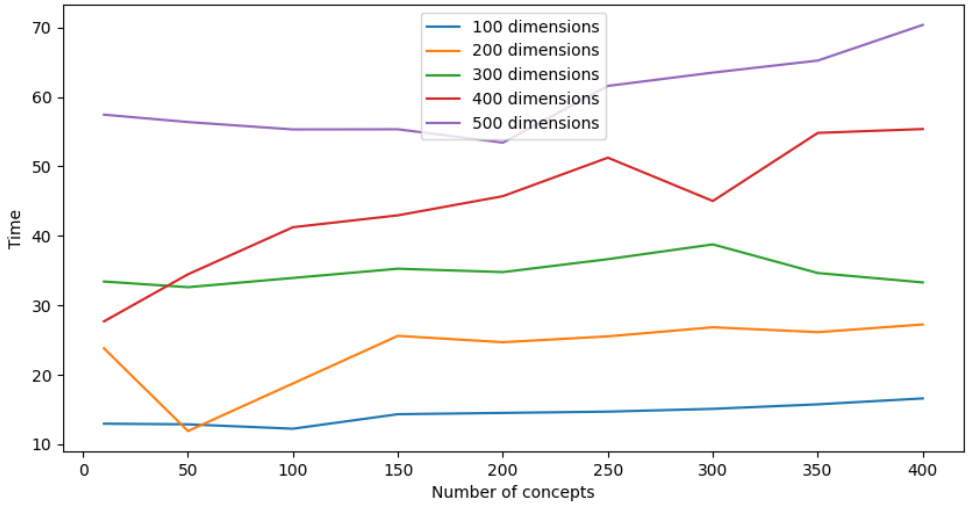


Figure 7: Computation time in seconds w.r.t number of embedding word dimensions and concepts.

From the experiment, we can see that the computation time increases as we increase the number of concepts or the number of word embedding dimensions. However, increasing the word embedding dimensions dramatically increases the calculation time compared to when increasing the number of concepts, as we can see in Figure (7).

8 Extension

The concept based method to represent documents can be used to address multiple use-cases inside an organisation. Possible extensions to this approach are applying text mining or machine learning models on top of this representation in order to extract useful information and / or to query documents based on the information they contain.

8.1 Clustering

Clustering can be used to group together documents that contains similar or related information. This will allow internal stakeholders within the organisation to refer to other documentations when searching for specific information. Like we discussed in Section (7.2), similar documents can be ranked according to their cosine distance with each other and the result can be shown to the user. Other approach can be to auto-tag documents based on the concepts they belong to. As we have seen that documents are a vector-representation of concepts where each concept consist

of similar or related tokens, these tokens can be used to tag documents for efficient retrieval of information. When uploading a new document, it's vector representation can be computed and then the corresponding top tokens from the most weighted concepts can be used to tag this new document.

Using the file - `PS Log Ingestion.docx` as an example with it's representation shown in Table (8), the following tags from the concepts that correlate to high weights in the representation could be considered.

Tag 1	Tag 2	Tag 3	Tag 4	Tag 5	Tag 6	Tag 7	Tag 8
decryption	kibana	scanner	storing	parser	archiver	debugger	download

8.2 Classification

Another extension to the proposed method is classification tasks of documents based on concepts. Such classification approaches could be used to perform narrow down search query of document retrieval. Classifying documents require a labelled dataset of documents that will be used to train a supervised learning algorithm. For example, given a use-case, documents can be labelled as "Study Proposal" or "Technical Report". The discussed method of representation of documents can be used to train a supervised learning algorithm such as a decision tree for this classification task. Following is an example of such decision tree.

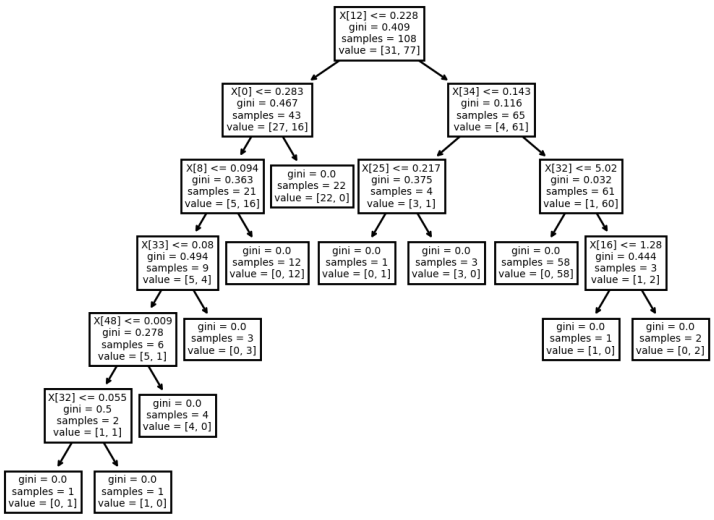


Figure 8: A Decision Tree

The concept based approach enables us to inspect the final model and intuitively understand how the model decides the final prediction. For example, we can clearly see the splitting criteria of each node in the decision tree that is based on the threshold of concept weights. Given a document with it's concept vector representation, we can check how the model processed this document in order to give out the final decision.

9 Conclusion

In this paper, we looked into various document representation techniques and proposed how a concept based approach can help analyse these document representations more intuitively. Furthermore, how the approach can help solve business challenges of identifying and analysing internal documents that are heterogeneous and covers multiple domains and knowledge areas. The paper also proposes how the relationship between the documents can be studied with respect to concepts. Even though we limit ourselves to few concepts for illustration purposes, we see that how the method can be extended to build more complex text mining and machine learning models on top of this representation.

References

[1] G.G. Chowdhury. *Introduction to Modern Information Retrieval*. Facet Publications. Facet, 2010. ISBN: 9781856046947. URL: https://books.google.se/books?id=cN4qDgAAQBAJ.

[2] Andrew M Dai, Christopher Olah, and Quoc V Le. "Document embedding with paragraph vectors". In: *arXiv preprint arXiv:1507.07998* (2015).

[3] Yoav Goldberg. *Neural Network Methods in Natural Language Processing*. Morgan Claypool Publishers, 2017. ISBN: 9781627052986.

[4] Derek Greene and Pádraig Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering". In: *Proc. 23rd International Conference on Machine learning (ICML'06)*. ACM Press, 2006, pp. 377–384.

[5] Zellig S. Harris. "Distributional Structure". In: *¡i¿WORD¡/i¿* 10.2-3 (1954), pp. 146–162.

[6] Han Kyul Kim, Hyunjoong Kim, and Sungzoon Cho. "Bag-of-concepts: Comprehending document representation through clustering words in distributed representation". In: *Neurocomputing* 266 (2017), pp. 336–352. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.05.046. URL: https://www.sciencedirect.com/science/article/pii/S0925231217308962.

[7] Elizabeth D. Liddy. *Document Retrieval, Automatic*. Encyclopedia of Language and Linguistics, 2005.

[8] Chao Xing et al. “Document classification with distributions of word vectors”. In: *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. 2014, pp. 1–5. DOI: 10.1109/APSIPA.2014.7041633.