Natural Language Processing Journal 8 (2024) 100081

Contents lists available at ScienceDirect

Natural Language Processing Journal

journal homepage: www.elsevier.com/locate/nlp

A modified Vector Space Model for semantic information retrieval

Callistus Ireneous Nakpih

C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana

ARTICLE INFO

Keywords:
Natural Language Processing
Vector Space Model
Information retrieval
Semantic retrieval
Cosine similarity

ABSTRACT

In this research, we present a modified Vector Space Model which focuses on the semantic relevance of words for retrieving documents. The modified VSM resolves the problem of the classical model performing only lexical matching of query terms to document terms for retrievals. This problem also restricts the classical model from retrieving documents that do not have exact match of query terms even if they are semantically relevant to the query. In the modified model, we introduced a Query Relevance Update technique, which pads the original query set with semantically relevant document terms for optimised semantic retrieval results. The modified model also includes a novel $tf-p$ which replaces the $tf-idf$ technique of the classical VSM, which is used to compute the Term Frequency weights. The replacement of the $tf-idf$ resolves the problem of the classical model penalising terms that occur across documents with the assumption that they are stop words, which in practice, there are usually such words which carry relevant semantic information for documents' retrieval. We also extended the cosine similarity function with a proportionality weight p_{qd}, which moderates biases for high frequency of terms in longer documents. The p_{qd} ensures that the frequency of query terms including the updated ones are accounted for in proportionality with documents size for the overall ranking of documents. The simulated results reveal that, the modified VSM does achieve semantic retrieval of documents beyond lexical matching of query and document terms.

1. Introduction

In the advent of advances in Natural Language Processing (NLP) research, several models have been developed for processing textual information with the core target of making natural language computable by machines, in order to make it possible for insight and meaning to be drawn from them in a more effective way (Khurana et al., 2023). These research efforts have been advanced to cover various linguistic areas such as; syntactic analysis, where models are developed to recognise the structure of sentences as well as the rules that govern their construction for various languages (Massa Cereda et al., 2018; Redd, 2014); semantic analysis, where models are developed to deduce or understand the meaning of text and their relatedness to other texts (Maulud et al., 2021; Pande and Karyakarte, 2019; Redd, 2014); morphological analysis, where models are developed to recognise root words and how other words are formed from them with affixes (Goyal and Lehal, 2008; Morphology Sphere, 2016; Vollmer, 2015); phonological analysis where models are developed to recognise the sound-patterns of a language and are able to translate them to text and vice versa (Covington et al., 1995; Shadiev et al., 2014). There are also other domain-specific areas, where NLP research seek to provide solutions targeting the processing of text which are unique to those domains; areas as varied as legal and philosophical semantics (Callistus, 2018; Nakpih and Santini, 2020), health informatics (Hao et al., 2021), social media analytics (Farzindar and Inkpen, 2017) and so forth.

The NLP models begun as rule-based techniques, where specific rules are explicitly defined for processing different linguistic features, which encompasses grammar rules, semantic rules, syntactic patterns etc. Given the complexity of natural language which is usually expressed in an unstructured form, combined with issues of ambiguities, as well as the diverse semantic and syntactic structure for thousands of languages that exist, it is near impossible to be able to explicitly define rules for all areas of linguistic features for different languages exhaustively. For this reason (among others), some mathematical, probabilistic and Machine Learning (ML) models or algorithms are used to process and to learn from existing textual data in order to create insight from them (Li and Shang, 2000; Nematzadeh et al., 2017; Wang et al., 2018). These algorithms are able to generate new text, classify text for different topics, identify linguistic features of language, translate text from one language to another, compare documents for similarity or retrieve information among other interesting processes.

One of the methods widely used for processing text is the Vector Space Model (VSM), which is a method used for representing text with some vectors in a high dimensional space. This method has been popularly used for Information Retrieval, Information Filtering, Text Translations etc., where queries and documents are vectorised, weighted and then computed via some algebraic similarity function, for ranking or retrieving documents that are closely related to the query terms (Singh, 2012).

E-mail address: cnakpih@cktutas.edu.gh.

https://doi.org/10.1016/j.nlp.2024.100081
Received 4 December 2023; Received in revised form 30 March 2024; Accepted 23 May 2024

The idea of VSM evolved over time, emerging from the classical definition of vector space, which asserts that, any vector can be obtained from other linear vectors (Dorier, 1995). The first operationalisation of VSM has been credited to Salton, who with other researchers provided various solutions and further advances for automatic document or information retrieval and dissemination (Salton, 1962, 1964, 1966, 1968; Salton et al., 1975). The operationalisation of the VSM went through several transformations which included some optimisation methods for retrieving documents.

Two key problems of the VSM continue to attract the attention of researchers; the first being that, the model is unable to retrieve documents based on their semantic relatedness to query terms, and the second being that, the model sometimes retrieves semantically irrelevant documents just because they contain some of the query terms. This phenomenon is made picturesque in the results of this research, where documents which have no semantic relation with queries were retrieved as the most relevant documents to their respective queries. Researchers continue to push the boundaries of these areas of VSM and other models that are used to present and to retrieve textual data, where the relevancy of retrieved documents is paramount to exact-match of terms in documents in this context. This compels research efforts to focus on how to include semantic information in the computation of the similarity of queries and documents, so as to improve on the areas of relevancy. We do agree that, measuring relevancy of documents can be a tricky concept in this domain, which has to be done with the mind of the appropriate query set, the target corpus, and even the people requiring the retrieved information. Having said that, there are several standard methods that suffice for evaluating the performance of the models in this regard (Sanderson, 2013).

On the score of improving relevancy of the VSM, we introduce several modifications to the classical VSM, which allows for the effective retrieval of lexically and semantically relevant documents.

2. Related works

Researchers have fairly pushed the frontiers of VSM, which has yielded some desirable innovations and improvement; having advanced from models that merely match exact query terms, to models that compute the probability of a document to be retrieved giving a query, as well as models that incorporates some semantic information for presenting and retrieving documents, are all remarkable advances in the VSM research domain. These advances by researchers include techniques such as Boolean Models, Generalised Vector Space Models, Latent Semantic Analysis, Word2Vec, GloVe, Machine Learning Models etc.

Early researchers in the VSM landscape including Lancaster and Frakes, modelled the presentation of text with Boolean principles for retrieving documents. These efforts mainly presented retrieval systems with exact lexical matching techniques which comes with some important challenges. The Boolean model is an algebraic method based on set theory, which uses decision tree techniques to record the presence or absence of a term in a document. The vector terms are recorded as 0 for the absence of a term in a document, and 1 for the presence of a term in a document; this record forms the Document Incidence Matrix. The Boolean operators AND, OR and NOT are then applied on the vectors of query terms to determine whether a document is relevant to a query set or not (Frakes and Baeza-Yates, 1992; Lancaster and Fayen, 1973; Lashkari et al., 2009).

The Boolean method (though classical) has been a long-standing technique that shaped information retrieval systems with some desirable results (reasonably). There are however a couple of problems that inherently come with the model. Notably among them are that; the Boolean model does not rank documents according the degree of their relevancy to a query. The documents are either retrieved or not, based on the application of the Boolean logic operations on the term vectors. Another critical challenge is that, the Boolean method only performs exact lexical matching of document terms to query terms. This results in the exclusion of some relevant documents which may have some degree of similarity to the query set. This exclusion is so because; on one hand, the meaning of words may change depending on the context they are used, and on the other hand, different words which may not necessarily share similar lexical features may be used to describe the same concept. Therefore, the ability to compare terms on the lexical level still does not guarantee similarity of terms, and consequently the non-retrieval of all relevant documents. Other challenges include, the Boolean model returning less relevant documents, or, retrieving too many documents which may not necessarily be relevant (Lashkari et al., 2009).

Some variants of the VSM were developed to resolve some of the problems found with the exact matching methods. The Generalised Vector Space Model (GVSM) for instance, which is an extension of the classical VSM, incorporates additional embedding of semantic data other than terms. The semantic relatedness of query terms and document terms are measured, by considering the relational weights and sense depth. This is done to improve the retrieval performance using the additional semantic information (Mumthaz et al., 2023; Tsatsaronis and Panagiotopoulou, 2009). Other VSM variants such as the Latent Semantic Analysis (LSA) also considers the semantic relatedness of terms by computing their co-occurrence in the same or similar text (Mumthaz et al., 2023; Dumais, 2004; Foltz, 1996). These models have successfully demonstrated that, techniques that include semantic information improves the computation of relevance of document to queries. This to some extent resolves the problems associated with methods for matching terms as a means of comparing documents.

Other VSM using Machine Learning techniques further advance the effectiveness of text presentation for information retrieval systems, even though they also pose their own challenges which include computational cost of implementing the models.

The Word2Vec model is one of the prevalent models that uses pre-trained neural-network word embeddings; two important learning models are used, viz, the Skip-gram and the Continuous Bag-of-words. The learning models are used to learn words that are close to a target word in order to establish the contextual use of words. Document terms are vectorised, and their semantic relatedness are then computed via the cosine function (Mikolov et al., 2013; Nematzadeh et al., 2017). As much as this model has been successful in learning and obtaining semantic and syntactic information of text, it is limited in several areas; the Word2Vec has no way of interpreting a new word it encounters in the course of prediction. This phenomenon is referred to as the out-of-vocabulary problem. Two other complex forms of limitation exist for Word2Vec; on one hand, the model discriminates between morphologically similar words, which in some instances should rather be treated as one and the same word, because they point to the same meaning but appear in different tenses or other forms of lexical structure that preserves the same meaning (homonymy) in a document. On another hand, a single word having different meaning (polysemy) in different context is represented by one vector which labels it with one meaning all the time. These among other challenges of the Word2Vec model causes it to miss out on some important semantic information of document terms.

Another important Machine Learning technique is the Global Vector (GloVe) (Pennington et al., 2014), which holds the assumption that, the distance between words presents some semantic information. The model therefore builds a vector presentation of words by computing the frequency of words that occur together. It uses the local context window-based (Collobert et al., 2011; Meng et al., 2020) and the global matrix factorisation (Lin and Pomerleano, 2013) methods for the learning of words. Since the local window-based method is unable to capture the statistical information of the global co-occurrences of words, the GloVe incorporated the matrix factorisation method to capture low-rank approximations which holds the global statistical information.

Even though GloVe and Word2Vec do present some effective ways of retrieving textual data, other researchers have criticised them, claiming that, the GloVe and the Word2Vec are not able to capture some words in the learning process. The argument is on the basis that, the intuitive similarity of words by human judgement is much complex and cannot be limited to distance in a vector space (Nematzadeh et al., 2017).

Other researchers have also presented a modified VSM targeted at retrieving documents based on synonyms and context of terms (Mumthaz et al., 2023). The research extends the query terms with synonyms whiles preserving the context of the queries and the documents. Individual synonym are weighted via a modified tf–idf weighting technique before they are passed through a similarity computation function for ranking and for improved document retrieval. The research presents a context-aware retrieval system compared to the classical VSM. However, The challenge (with regards to semantic retrieval) with this approach is that, there are words that are semantically related but are not directly synonyms. This in effect will also cause the approached to miss out on some semantically relevant words for the retrieval processes. It is also not clear how the $tf - idf$ was modified and the effect it has on the whole retrieval process.

While some research works (Naol, 2019) implement the classical VSM with modification effort on the pre-processing techniques which are enhanced to improve the performance of the classical VSM; including the incorporation of a thesaurus for the VSM to access synonym terms to form part of the retrieval process, others (Wintana et al., 2019) have also implement the VSM for some domain specific information retrieval, while others implement the VSM for specific file systems (Wahyudi et al., 2019). Invariably, the classical VSM has had different target research areas, all of which are tended towards enhancing it for improved retrieval of documents.

3. Methodology

In this research we selected 12 documents as a testbed to illustrate the implementation of the proposed model, as well as its performance against the classical model. The testbed documents include articles on Ghana, United States, Politics, Computer Science, Mathematics, Finance, Human Body, Human Chimera, Enzyme, Ghana Banking Crisis, Ghana Economic Turmoil, World Financial Crisis and its implications for Ghana. All the documents are 2023 Wikipedia documents except *Ghana Economic Turmoil* which is a news article, and *World Financial Crisis and its implications for Ghana* which is a parliamentary paper (note that the title of this document has been revised to *World financial crisis-gh* in the simulation for convenience). These documents were particularly selected because they can be categorised into identifiable themes; some of them have closely related vocabulary with each other, which presents the right scenario for the proposed and classical models to be tested to ascertain their degree of discrimination between documents that are of different degree of relevancy to a query. Another key purpose these documents serve is, to ascertain whether the models are able to recognise documents that do not contain exact match of a query term, but are semantically related to the query. The small corpus choice and simulation setup is purposely done to allow for the illustration and discussion of the rudimentary semantic retrieval concepts and capabilities of the proposed model.

3.1. Data pre-processing

The data pre-processing techniques used in this research was done to get textual data in a form that is devoid of noise, and for further processes or analysis. The text in the documents included punctuations, mixed character formats from different natural languages, non-word strings, metadata and so forth. The pre-processes stage primarily ensured that the text is tokenised, and is cleaned by removing all noise and stop words from it.

One important feature of the proposed model is the replacement of the $tf - idf$ of the classical VSM, which penalises more frequent terms across documents, which we argue that, not all frequently occurring terms in documents are stop words, some may actually carry some relevant semantic information for a document. However, the $tf - idf$ technique has no way of discriminating between actual stop words and non-stop-words. Therefore, in order to maintain the use of only relevant terms, and at the same time excluding the actual stop words and other irrelevant terms from the documents for this research, we adopted a data cleaning system that removes all actual stop words as well as punctuations, emails, URLs, non-ASCII characters, numeric values, contractions, concurrency symbols etc.

Pre-processing techniques like stemming and lemmatization of words change the lexical and morphological form of words, which sometimes changes the semantic value of the words, or sometimes generate words that do not make sense. Having words in a lexical or morphological form that maintains their semantic meaning is very crucial for the modified VSM, for which reason lemmatisation and stemming are excluded in the pre-processing steps for this research.

3.2. The theoretical framework of the classical vector space model

The proposed model stems out of the classical VSM framework. The classical VSM generally computes the Term Frequencies tf, which is the number of times a term occurs in a document; the Document Frequency df, which is the number of documents containing terms; and the Inverse Document Frequency idf, which is computed by dividing the total number of documents by the df. The log of idf is used to compute the weights of the query term vectors by multiplying $\log(idf)$ by each tf of the documents. This mechanism is generally referred to as the tf–idf scheme, which is used by most IR systems as a weighting technique to determine how a term is important to a document.

There are different forms of tf which are used for different IR systems. However, the basic form of it is formally defined as $tf(w_i, d_j)$, where w_i are terms in documents d_j. In other words, the functions $tf()$ counts the occurrences of w_i in d_j. The idf is as well formally defined as;

$idf\left(w_i, D\right) = \log\left(\frac{|D|}{\left|\left\{d_j \in D : w_i \in D\right\}\right|}\right)$, where;

$|D|$: total number of documents

$|\left\{d_j \in D : w_i \in D\right\}|$: number of documents containing terms.

The $tf - idf$ is therefore expressed as follows;

$$tf - idf(w_i, d_j, D) = tf\left(w_i, d_j\right) * idf(w_i, D)$$

The $tf - idf$ are the final weights that are then passed to the cosine similarity function for computing the similarity of queries and documents.

$$\text{Cosine similarity} = \frac{\sum_{i=1}^{n} w_i d_j}{\sqrt{\sum_{i=1}^{n} w_i^2}\sqrt{\sum_{i=1}^{n} d_j^2}}$$

3.3. Multiset theory

The proposed model is presented using the Multiset Theory (Singh et al., 2007), and algebraic expressions. The Multiset Theory was used as the model language because, it provides a much simpler presentation of count functions mathematically, using the multiplicity of elements (even under conditional constraints) as well as the presentation of cardinality more conveniently. It also allows for a more convenient definition of the document and query terms to be presented as repeatable elements, unlike in the classical set theory, where elements are considered non-repeating. The number of times an element appears in a multiset is known as the *multiplicity* of the elements, and the sum of multiplicity is known as the *cardinality*.

4. The modified vector space model

The proposed model modifies several aspects of the VSM approach. First, a technique for updating a query set is introduced, where the original query set is padded with closely semantically related terms from documents in the search corpus. The updated query set is then used to generate the Term Frequencies for all documents, that is, the number of occurrences of terms in the documents are counted and recorded as decimal values. Note that, the Term Frequencies of the query set is done by generating binary values, where 1 represents a term being a query term and 0 otherwise.

Second, the idf is replaced with the proportionality of the sum of occurrences of document terms to the document size, iff they are query terms; we denote this proportionality as p. In effect, we introduced a $tf - p$ technique in place of the $tf - idf$, which is the computed Term Frequency Weights.

Third, a query size to document size proportionality p_{qd} is also computed, which is used to extend the cosine similarity function to regulate the effect of document length on the final ranking of documents. This is done to prevent higher frequencies in longer documents from having an undue or false advantage over shorter documents. The proportionality mechanism ensures that, the document ranking is contingent on the frequencies of terms in proportion to the document size.

Therefore, we define d_λ for documents containing terms w_i, and a set of original queries q as follows;

$$d_\lambda = \left\{ w_{i,\lambda} | w_{i,\lambda} \in d_\lambda, for\ some\ \lambda \right\},$$

$$q = \left\{ w_{i,q} \middle| w_{i,q} \in q \right\}$$

Given the identity I, such that; $I = \{x | 1 \le x \le n\}$,

We denote $D = \bigcup_{\lambda \in I} d_\lambda$ where d_λ is an element of D; in other words, d_λ is the individual documents from which we search for query terms, and D is the total number of documents.

4.1. Query relevance update

The updated query set q' is generated to include highly similar document terms in the query set; we term this technique as *Query Relevance Update*. The idea of q' is slightly similar to that of the Pseudo Relevance Feedback (PRF) (Chen et al., 2022), where the initial query terms are used to rank documents. The technique assumes that, high ranking documents are relevant to the user, and therefore pads the query set, or reweights the query terms with terms from the high-ranking documents and then runs a search again for a better ranking of documents.

However, for the q' in our model (as differentiated from the PRF), the original query terms are assessed for their semantic similarity with each term in the documents. The query set is then automatically updated with the document terms that have high similarity with the original query terms. The similarity at this stage is computed using word vectors and the multi-dimensional meaning representation of terms. The updated query set is then used as the new query set for the rest of the computation in the model for ranking documents. Unlike the PRF technique, our model only performs one ranking at the final stage using the generated q'.

We set a threshold of 0.65 similarity between $w_{i,q}$ and $w_{i,\lambda}$; if the similarity is 0.65 or greater, the document term will be added to the query set as a query term. Our simulations reveal that, a very high threshold may exclude some important terms, while very low threshold may also include irrelevant terms. At the moment, we have no technical way of picking the best threshold for the proposed model. However, the 0.65 threshold as shown in the results has proven to give very good results so far. The updated query set is therefore defined as follows;

We define a binary relation on d_λ and q by " = " as

$${}_{w_{i,q}}R_{w_{i,\lambda}} = \left\{ (w_{i,q}, w_{i,\lambda}) \in qXd_\lambda \middle| w_{i,q} = w_{i,\lambda} \right\},$$

The relation R is symmetric, which denotes the mechanism for retaining original query terms that are exactly the same as terms in the documents. This part of the model is just to ensure that, the exact-matching terms are not excluded from the updated query set. This relation is therefore extended to compose the final updated query set q' with documents terms that are not exactly the same as query terms but have high similarity. This is done as follows;

$$q' = \left\{ w_{i,q'}, w_{i,\lambda} \middle| w_{i,q'} \in q, and\ sim\left(w_{i,q}, w_{i,\lambda}\right) \ge 0.65 \right\}$$

After obtaining q', we compute the binary vectors of the terms in the query set q'_v as follows; 1 is recorded for a term if it is a query term (including the added document terms), and 0 if it is not.

$$q'_v = \begin{cases} 1, & if\ w_{i,q} \in q' \\ 0, & otherwise \end{cases}$$

4.2. Term frequencies

We define a multiset function $f_{d_\lambda}\left(w_{i,q'}\right)$ for $w_{i,q} \in^n d_\lambda$, where $w_{i,q}$ is an element of d_λ with a multiplicity of n.

$$f_{d_\lambda}\left(w_{i,q'}\right) = \begin{cases} n, & if\ w_{i,q'} \in^n d_\lambda \\ 0, & otherwise \end{cases}$$

The function $f_{d_\lambda}\left(w_{i,q'}\right)$ counts the frequencies for the terms $w_{i,q}$ in each document d_λ.

For instance, if $\lambda = 1$, and $i = 1$. We imply that, term 1 (w_1) in document 1 (d_1) is counted via the function $f_{d_\lambda}\left(w_{i,q'}\right)$, and its multiplicity is recorded. Therefore, if w_1 occurs 7 times in d_1, then $n = 7$ will be recorded as the Term Frequency for w_1 in d_1, and so forth. The tf is then normalised as follows;

$$tf_{norm} = \frac{tf - Min(tf)}{MAX\,(tf) - MIN(tf)}$$

The normalisation is done to keep the tf within a manageable range of values and also to avoid data explosion, since there can be thousands of generated terms and documents as well. Fig. 1 illustrates how the data is presented in a matrix format.

4.3. Cardinality of terms and proportionality weights

We compute the sum of vectors for all $w_{i,q'}$ for each d_λ. This is the computation of the cardinality (sum of multiplicity) of all frequencies in each document d_λ, with the function $card\left(d_\lambda\right)$. So, for all query terms $w_1, \ldots\ldots, w_k$, we sum up the frequencies for each document, d_1, d_2, $d_3, \ldots, d_\lambda$. This is done as follows;

$$card\left(d_\lambda\right) = \sum_{i=0}^{k} f_{d_\lambda}\left(w_{i,q'}\right)$$

We also introduce a Conditional Cardinality function $Con_{card(d_\lambda)}$, where the conditional sum of frequencies of query terms in documents d_λ are computed; for instance, in document 1 (d_1), only frequencies of query terms occurring in d_1 are summed up. This is different from the sum of frequencies of all terms that occur in d_1. The conditional sum or cardinality is therefore expressed as;

$$Con_{card(d_\lambda)} = \begin{cases} \sum_{i=0}^{k} f_{d_\lambda}\left(w_{i,q'}\right), & if\ q'_v\left(w_{i,q'}\right) = 1 \\ 0, & otherwise \end{cases}$$

For a every Term Frequency in a document, its corresponding query vector is checked, if the q'_v is 1, implying that the term is a query term, the Term Frequency in the document will be part of the conditional sum.

A Conditional Cardinality to the documents size proportionality p is introduced, which is used to compute the Term Frequency Weights.

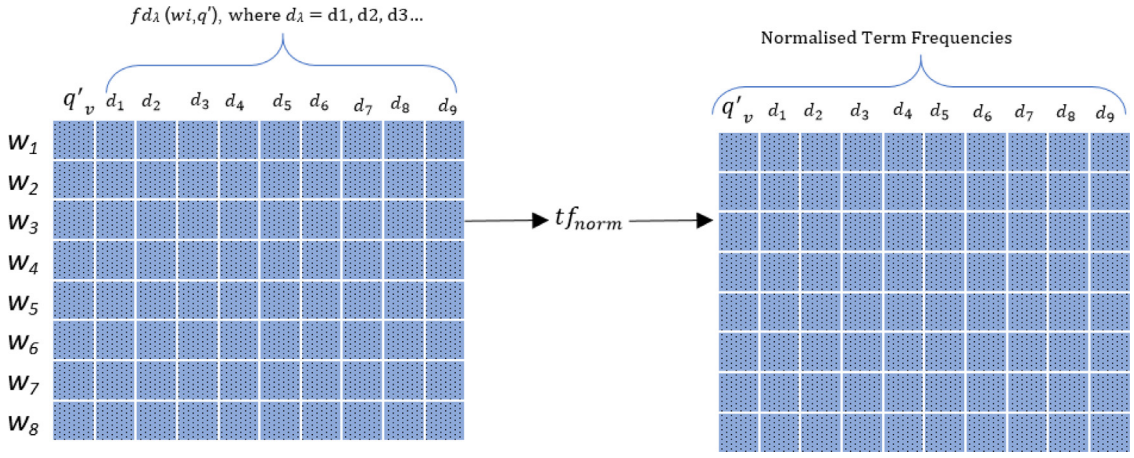

Fig. 1. Normalising query and document term frequencies.

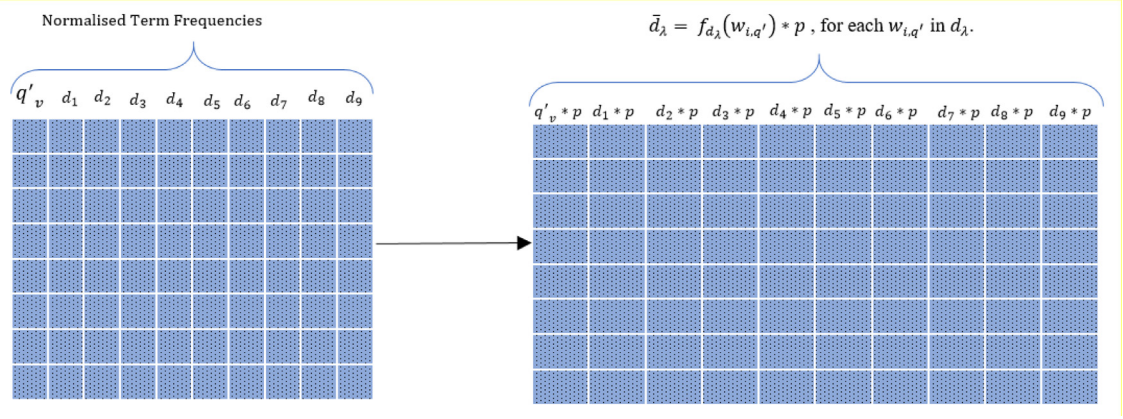

Fig. 2. Computing $\bar{d}_\lambda$ from the normalised term frequencies.

This weighting system accounts for the frequency of query terms in a document relative to a document size.

$$p = \frac{con_{card(d_\lambda)}}{card\,(d_\lambda)}$$

The Term Frequency Weights for the modified model are computed by multiplying each Term Frequency by p (the p replaces the $log(idf)$ in the classical VSM). The following expression generates the Term Frequency Weights for all documents;

$tf{-}p = tf * p$, we represent $tf - p$ by $\bar{d}_\lambda$, and tf by $f_{d_\lambda}(w_{i,q'})$, therefore;

$\bar{d}_\lambda = f_{d_\lambda}(w_{i,q'}) * p$ for each $w_{i,q'}$ in d_λ. Fig. 2 illustrates the matrix form of how the normalised Term Frequencies are weighted by multiplying each tf by p. The weighted tf, thus, are the final values that will be passed to the extended cosine computation for the ranking of documents.

4.4. Extended cosine function

The p_{qd} is the proportionality of the query size to the document size. Since the query set is updated with relevant terms from the documents, the new query terms increase the query size as well as the tf, especially for documents with more semantically related terms. This can introduce a bias for longer documents just because they have a larger set of vocabulary. This proportionality is therefore used to extend the cosine function, in order to moderate the effects of the term frequencies on the final ranking based on document size. This is expressed as follows;

$$p_{qd} = \frac{\sum_{i=0}^{k} q'_v}{card\,(d_\lambda)}\text{ , for each document } d_\lambda,$$

for each document d_λ

Since a query term is represented by 1 and non-query terms by 0, the sum of q'_v is the same as the query size. The cosine function takes the set $\bar{d}_\lambda$, which is the set of Term Frequency Weight and q'_v as arguments; the extension is particularly done by multiplying the dot product by the p_{qd}. The extended cosine similarity function is expressed as follows;

$$cosine\left(\bar{d}_\lambda, q'_v\right) = \frac{\left(\sum \bar{d}_\lambda * q'_v\right) * p_{qd}}{\sqrt{\sum(\bar{d}_\lambda)^2 * \sum(q'_v)^2}}$$

The following algorithm framework presents the complete process of the modified VSM.

Table 1
Statistical Information and Proportionality Weights of Testbed Documents.

	Ghana	US	Politics	Chimera	Computer	Math	Enzyme	Finance	Ghana Bank crisis	Ghana economic turmoil	world financial crisis-gh	Human body
MIN	0	0	0	0	0	0	0	0	0	0	0	0
MAX	193	260	124	45	139	325	171	164	75	34	35	95
$card\ (d_\lambda)$	4754	7862	2997	1592	2846	9106	6107	3944	1377	1874	1900	3901
$Con_{card(d_\lambda)}$	271	60	17	5	3	22	22	424	93	135	173	28
p	0.05700	0.00763	0.00567	0.00314	0.00105	0.00242	0.00360	0.10751	0.14016	0.07204	0.09105	0.00718
p_{qd}	0.00926	0.00560	0.01468	0.02764	0.01546	0.00483	0.00720	0.01116	0.03195	0.02348	0.02316	0.01128

Algorithm 1: The Modified VSM

1. Start
2. Read Query and document inputs: q, d_λ
3. Pre-processing q, d_λ: Tokenise, Remove noise, Remove stop words
4. If $sim\ (w_{i,q}, w_{i,\lambda}) \geq 0.65$
5. $q' \leftarrow w_{i,\lambda}$
6. Generate query tf: $q'_v = \begin{cases} 1, & if\ w_{i,q} \in q' \\ 0, & otherwise \end{cases}$
7. Generate document tf: $f_{d_\lambda}\ (w_{i,q'}) = \begin{cases} n, & if\ w_{i,q'} \in^n d_\lambda \\ 0, & otherwise \end{cases}$
8. Normalise query and document tf: $tf_{norm} = \frac{tf - Min(tf)}{MAX(tf) - MIN(tf)}$
9. Compute: $card\ (d_\lambda) = \sum_{i=0}^{k} f_{d_\lambda}\ (w_{i,q'})$
10. Compute: $Con_{card(d_\lambda)} = \begin{cases} \sum_{i=0}^{k} f_{d_\lambda}\ (w_{i,q'}), & if\ q'_v\ (w_{i,q'}) = 1 \\ 0, & otherwise \end{cases}$
11. Compute: $p = \frac{con_{card(d_\lambda)}}{card(d_\lambda)}$
12. Compute tf–p: $\overline{d}_\lambda = f_{d_\lambda}\ (w_{i,q'}) * p$
13. Compute: $p_{qd} = \frac{\sum_{i=0}^{k} q'_v}{card(d_\lambda)}$
14. Compute extended cosine similarity for ranking:
$cosine\ (\overline{d}_\lambda, q'_v) = \frac{\left(\sum \overline{d}_\lambda * q'_v\right) * p_{qd}}{\sqrt{\sum(\overline{d}_\lambda)^2 * \sum(q'_v)^2}}$
15. End

5. Simulated results

The proposed model was implemented and evaluated for its ability to retrieve semantically related documents to query terms. The evaluation reveals that, our $tf - p$ technique preserves the relevance of non-stop-words for the retrieval of documents in the frame of the document size without a penalisation effect.

The Table 1 presents the key statistical information generated from the cleaned testbed documents based on the query, *what caused the financial crisis in Ghana?*. The table includes the computed p and p_{qd} weights for all document. The MAX and MIN information were used to compute the tf_{norm} while the $card(d_\lambda)$ and $con_{card}(d_\lambda)$ were used to compute p and p_{qd}. The p weights were used to compute the term frequency weights, and the p_{qd} was used to extend the dot product of the cosine similarity function.

In Table 1, The MIN row records the minimum Term Frequency, which is 0 for each document, while the MAX row records the maximum Term Frequency for each document. The value 193 recorded as MAX value for the *Ghana* document means that there is a term that occurs 193 times in that document, which is the highest frequency among all the Term Frequencies in the document *Ghana*. The Cardinality values are computed by summing up all the Term Frequency for each document. The value 4754 for the document *Ghana* is the sum of frequencies for all terms occurring in that document. The Conditional Cardinality however, sums up only frequencies of document terms that are query terms. That is why the value 271 which is the computed $Con_{card(d_\lambda)}$ is far smaller than the value for the $card\ (d_\lambda)$ for the document *Ghana*. The p values are computed by dividing the $Con_{card(d_\lambda)}$ by the $card\ (d_\lambda)$; in the case of the *Ghana* document, it is $\frac{271}{4754}$ which gives us 0.05700. The p values are also used to compute the $\overline{d}_\lambda$ or tf–p weights, by multiplying the p by each Term Frequency for each document, which gives us $\overline{d}_\lambda$ (weighted values). These weighted values of the terms in each document become the arguments for the extended cosine similarity function. The computed cosine values are used to rank the documents, where higher cosine values mean strong similarity between a query and a document, and lower cosine values mean weak or no similarity between the query and a document.

For this simulation and evaluation, the proposed model is used to rank the documents based on the query, *"what caused the financial crisis in Ghana?"*. This ranking is then compared to the ranking of the classical VSM as presented in Table 2. The original query size was 7, but it increased to 44 after it was updated with closely semantically related terms from the documents. Note that, the updated query set is only used by the modified model.

The query was targeted at retrieving documents that have information on the financial crisis in Ghana. Four keywords of the original query set played a major role in the document retrieval process; *financial, ghana, crisis,* and *caused*. These key words were carefully used as the query terms because, their occurrences in the documents offer a unique opportunity for analysing the relevance of retrieved documents.

The document *Ghana* in our small corpus, is an article that generally describes the nation and has little to do with the query information. However, as illustrated in Table 2, it is ranked number 1 by the classical model, which is undesirable, and one of the typical problems of the classical VSM, where irrelevant documents are retrieved for a query. The Fig. 3 shows how the query terms are distributed across the documents, from which it is very picturesque that, the highest query Term Frequency is in the document *Ghana* which is what is accounting for that document to be considered as the most relevant to the query, while in reality, the document *Ghana* is not relevant. The query is not just requesting for a document with information about Ghana (or with terms *ghana*). It is requesting for documents with information on the financial crisis of Ghana. On that account, the modified model has proven to generate better results by ranking high, more relevant documents per the query terms, and rather ranks the document *Ghana* to be 5.

The modified model does not just consider documents with high frequencies of terms, or, documents with single dominating terms to be relevant. For instance, the article *Ghana Bank crisis*, does not have as high frequencies as other documents as seen in Fig. 3, however, from Table 2, we see that it is ranked 1 by the modified model, and this is because the model is able to consider other document terms which are relevant to the query but are not lexically the same as the query terms, and therefore gives such documents high ranks because of their close semantic relation to the query term. The documents ranked from 1 to 4 by the modified model are relevant to the query text per their content. A manual look at the content of the documents also confirms that the modified model is indeed achieving a semantic retrieval of documents as opposed to the classical model.

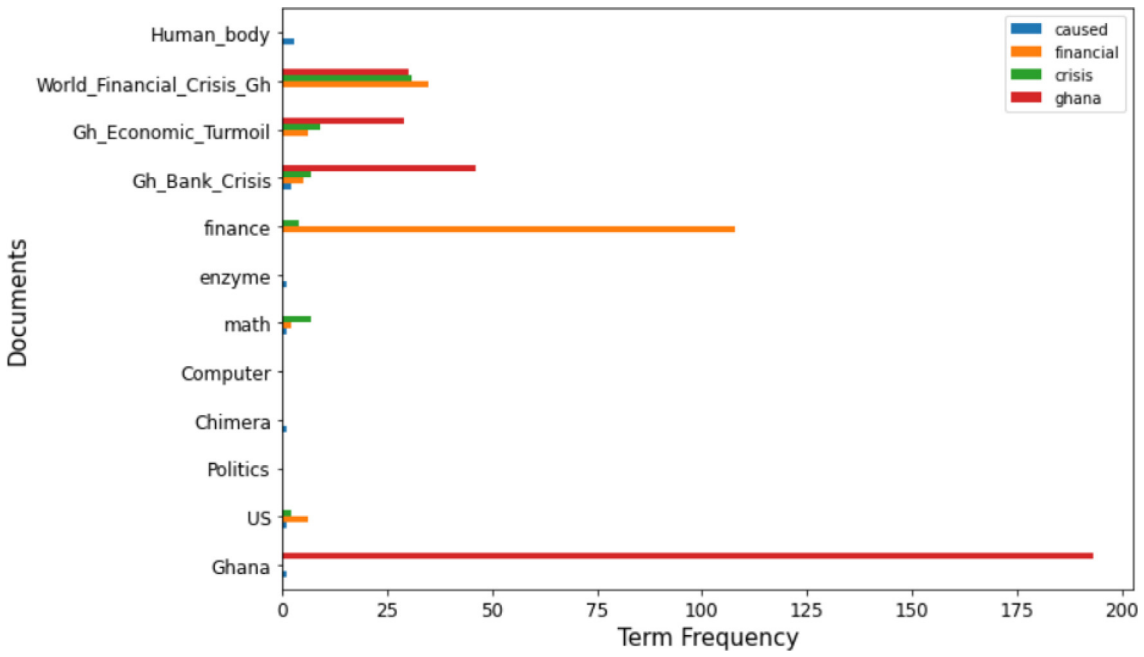

Fig. 3. Query term frequencies in documents.

Table 2
Document Ranking for the query; *what caused the financial crisis in Ghana.*

Document	Modified VSM Ranking	Classical VSM Ranking
Ghana Bank crisis	1	4
world fin crisis-gh	2	2
Ghana economic turmoil	3	3
finance	4	5
Ghana	5	1
Chimera	6	10
Human body	7	7
Politics	8	8
US	9	6
enzyme	10	11
Computer	11	12
math	12	9

Table 3
Document Ranking for the Query; *what are the functions of micro organisms in the stomach?*

Document	Modified VSM Ranking	Classical VSM Ranking
Human body	1	2
Chimera	2	10
enzyme	3	9
world fin crisis-gh	4	4
Ghana economic turmoil	5	7
finance	6	8
Politics	7	5
Computer	8	11
US	9	1
math	10	6
Ghana	11	3
Ghana Bank crisis	12	12

To further emphasise the semantic retrieval capabilities of the modified VSM against the classical VSM, we tested the two models with a single query term "*Calculator*". This is done to illustrate the effect of the presence or absence of a query term in a document. The term used for this test is particularly done because, it is only present in the document *computer* and absent in the rest. This consequences in cosine values for documents other than *computer* to be 0, and in essence makes those documents non-retrievable by the classical model. In Fig. 4, only *computer* is seen to be retrievable by the classical VSM, because it is the only document with the query term.

In contrast to the behaviour of the classical VSM, as shown in Fig. 5, the documents *computer*, *math* and *Ghana economic turmoil,* even though do not contain the query term have cosine values greater than 0, and hence retrievable by the modified model. This is because, the modified model is able to relate to the documents without the exact lexical match of the query term via the semantically relevant terms for the retrieval process. These three documents have different degree of contents encompassing the idea of mathematical calculations or computations, hence, the reason for their retrievability.

The Table 3 also emphasises the semantic capabilities of the modified VSM against the classical VSM. The modified VSM ranks up all relevant documents containing biological information in the corpus from 1 to 3, which are closely related to the query. The classical VSM on the other hand ranks up documents that are not related to the query. The document *US* and *Ghana* which are not related to the query are ranked 1 and 3 respectively by the classical model. This is because these documents merely contain lexical matches of some of the query terms in them. Based on these results, it is clear that the modified VSM has proven to go beyond the lexical comparison of query and document terms. Its semantic retrieval capabilities is an important step for information retrieval systems where relevancy of documents is much desired than retrieval of documents with exact query terms.

6. Conclusion

This research targeted modifying the classical VSM to achieve semantic retrieval capabilities. The key function of VSM is to find how close a query is to a document in a vector space, which is largely determined by the query Term Frequencies in documents. This analysis in the classical VSM is done at the lexical level, and therefore relates a query to a document based on their lexical similarity. This makes the classical VSM miss out on some relevant documents which may not have the same query terms. On that score, we presented a model that includes techniques that allow for the recognition, or, retrieval of documents that are related to a query semantically. We therefore, chose some testbed documents in order to illustrate the semantic retrieval concepts of the models.

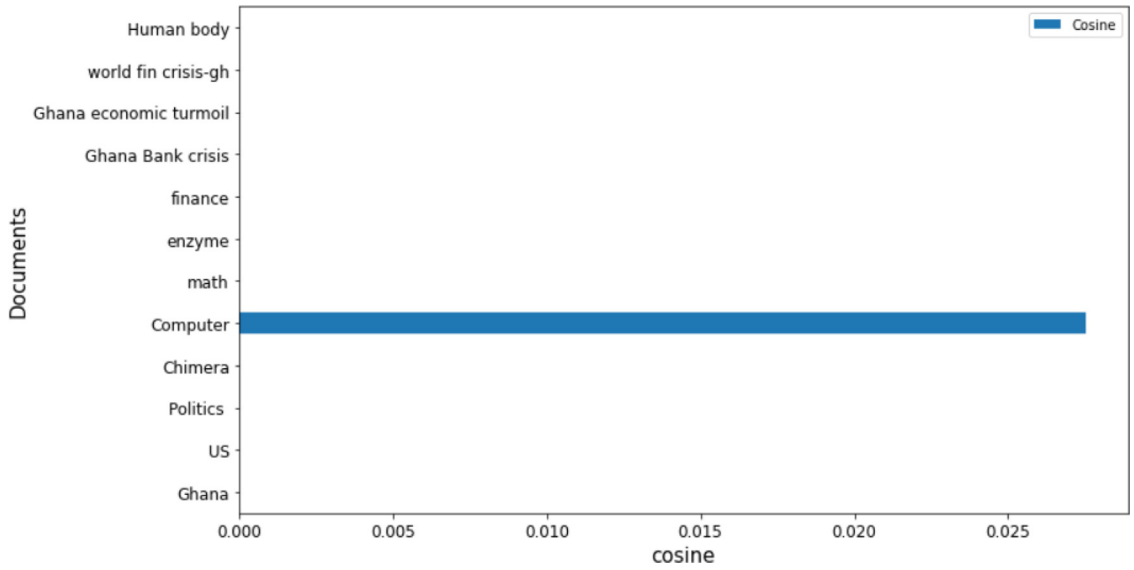

Fig. 4. Cosine of classical VSM.

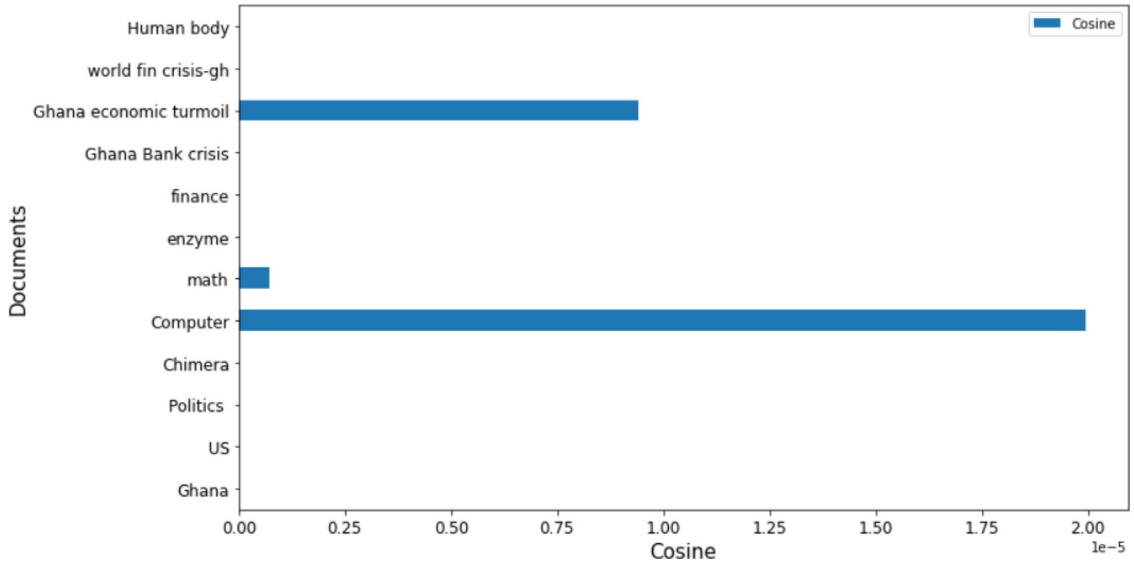

Fig. 5. Cosine of modified VSM.

In our model, we replaced the $tf - idf$ technique with a novel $tf - p$ approach for the computation of the Term Frequency Weights; this resolves the problem of terms being penalised for occurring across documents; this penalisation effect diminishes the weights of such terms, and consequently make them have little or no effect on the overall ranking of documents with the assumption that they are stop words. However, if such terms are not necessarily stop words, then their weights should be significant for the ranking of documents, which the classical model has no way of establishing. The $tf - p$ eliminates this penalisation effect, while also avoiding the use of stop words as part of the data via a rigorous pre-processing step which removes them. The document size is accounted for in the computation of the $tf - p$ and in the extension of the cosine similarity function; this is done to disallow the size of a document from being a bias influence on its retrieval.

The simulation illustrated how high Term Frequencies at the lexical level do not necessarily influence high ranking of a document in the modified model. The modified model is able to recognise documents that are semantically related to a query, and can disregard documents that are seemingly relevant to a query (seemingly because of high occurrence of some query term in them). For instance, we see in the simulation that, the document *Ghana* which has little to do with the *financial crisis of Ghana* was ranked low by the modified model while the classical model ranks it as the highest relevant document in the corpus. We conclude that, the modified model as demonstrated, achieves semantic retrieval and offers more semantic related documents to a query than the classical VSM.

CRediT authorship contribution statement

Callistus Ireneous Nakpih: Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Callistus, I.N., 2018. Citizenship act (591) of Ghana as a logic theorem and its semantic implications. Int. J. Nat. Lang. Comput. 7 (5), 11–25. http://dx.doi.org/10.5121/ijnlc.2018.7502.

Chen, Z., Guo, N., Sun, J., Wang, Y., Zhou, F., Xu, S., Wang, R., 2022. Pseudo-relevance feedback method based on the topic relevance model. Math. Probl. Eng. 2022, e1697950. http://dx.doi.org/10.1155/2022/1697950.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural language processing (almost) from scratch. http://dx.doi.org/10.48550/arXiv.1103.0398, arXiv. (arXiv:1103.0398).

Covington, M.A., Nute, D., Vellino, A., 1995. Prolog programming in depth.

Dorier, J.-L., 1995. A general outline of the genesis of vector space theory. Historia Math. 22 (3), 227–261. http://dx.doi.org/10.1006/hmat.1995.1024.

Dumais, S.T., 2004. Latent semantic analysis. In: Annual Review of Information Science and Technology (ARIST). Vol. 38, pp. 189–230.

Farzindar, A., Inkpen, D., 2017. Natural Language Processing for Social Media, second ed. In: Synthesis Lectures on Human Language Technologies, pp. 101–195. http://dx.doi.org/10.2200/S00809ED2V01Y201710HLT038.

Foltz, P.W., 1996. Latent semantic analysis for text-based research. Behav. Res. Methods Instrum. Comput. 28 (2), 197–202. http://dx.doi.org/10.3758/BF03204765.

Frakes, W.B., Baeza-Yates, R. (Eds.), 1992. Information Retrieval: Data Structures and Algorithms. Prentice-Hall, Inc.

Goyal, V., Lehal, G.S., 2008. Hindi morphological analyzer and generator. In: 2008 First International Conference on Emerging Trends in Engineering and Technology. pp. 1156–1159. http://dx.doi.org/10.1109/ICETET.2008.11.

Hao, T., Huang, Z., Liang, L., Weng, H., Tang, B., 2021. Health natural language processing: Methodology development and applications. JMIR Med. Inf. 9 (10), e23898. http://dx.doi.org/10.2196/23898.

Khurana, D., Koli, A., Khatter, K., Singh, S., 2023. Natural language processing: State of the art, current trends and challenges. Multimedia Tools Appl. 82 (3), 3713–3744. http://dx.doi.org/10.1007/s11042-022-13428-4.

Lancaster, Frederick Wilfrid, Fayen, Emily Gallup, 1973. Information Retrieval: On-Line. Melville Publishing Company.

Lashkari, A.H., Mahdavi, F., Ghomi, V., 2009. A boolean model in information retrieval for search engines. In: 2009 International Conference on Information Management and Engineering. pp. 385–389. http://dx.doi.org/10.1109/ICIME.2009.101.

Li, L., Shang, Y., 2000. A new statistical method for performance evaluation of search engines. In: Proceedings 12th IEEE Internationals Conference on Tools with Artificial Intelligence. ICTAI 2000, pp. 208–215. http://dx.doi.org/10.1109/TAI.2000.889872.

Lin, K.H., Pomerleano, D., 2013. Global matrix factorizations. http://dx.doi.org/10.48550/arXiv.1101.5847, arXiv. (arXiv:1101.5847).

Massa Cereda, P.R., Miura, N.K., Neto, J.J., 2018. Syntactic analysis of natural language sentences based on rewriting systems and adaptivity. Procedia Comput. Sci. 130, 1102–1107. http://dx.doi.org/10.1016/j.procs.2018.04.164.

Maulud, D., Zeebaree, S., Jacksi, K., M. Sadeeq, M., Hussein, K., 2021. A state of art for semantic analysis of natural language processing. Qubahan Acad. J. 1, http://dx.doi.org/10.48161/qaj.v1n2a44.

Meng, Y., Huang, J., Wang, G., Wang, Z., Zhang, C., Han, J., 2020. Unsupervised word embedding learning by incorporating local and global contexts. Front. Big Data 3, https://www.frontiersin.org/articles/10.3389/fdata.2020.00009.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv. (arXiv:1301.3781). http://arxiv.org/abs/1301.3781.

Morphology Sphere, 2016. Natural Language Processing and Computational Linguistics. Vol. 1. John Wiley & Sons, Ltd, pp. 89–125. http://dx.doi.org/10.1002/9781119145554.ch3.

Mumthaz, B., S, A., Vijayan, R., 2023. Synonym insensitive searching: a novel synonym weighted-vector space model for document retrieval. In: 2023 2nd International Conference on Computational Systems and Communication. ICCSC, pp. 1–7. http://dx.doi.org/10.1109/ICCSC56913.2023.10142977.

Nakpih, C.I., Santini, S., 2020. Automated discovery of logical fallacies in legal argumentation. Int. J. Artif. Intell. Appl. 11 (2), 37–48. http://dx.doi.org/10.5121/ijaia.2020.11203.

Naol, Bakala, 2019. Information retrieval system by using vector space model. Int. J. Sci. Technol. Res. 8 (12), 1562–1568.

Nematzadeh, A., Meylan, S.C., Griffiths, T.L., 2017. Evaluating vector-space models of word representation, or, the unreasonable effectiveness of counting words near other words.

Pande, N., Karyakarte, M., 2019. A review for semantic analysis and text document annotation using natural language processing techniques. http://dx.doi.org/10.2139/ssrn.3418747, (SSRN Scholarly Paper 3418747).

Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. EMNLP, pp. 1532–1543. http://dx.doi.org/10.3115/v1/D14-1162.

Redd, M.V., 2014. Semantical and Syntactical Analysis of NLP. Vol. 5.

Salton, G., 1962. Some experiments in the generation of word and document associations. In: Proceedings of the December (1962) 4-6, Fall Joint Computer Conference. pp. 234–250. http://dx.doi.org/10.1145/1461518.1461544.

Salton, G., 1964. A document retrieval system for man-machine interaction. In: Proceedings of the 1964 19th ACM National Conference On -, 122.301-122.3020. http://dx.doi.org/10.1145/800257.808923.

Salton, G., 1966. Data manipulation and programming problems in automatic information retrieval. Commun. ACM 9 (3), 204–210. http://dx.doi.org/10.1145/365230.365267.

Salton, G., 1968. Search strategy and the optimization of retrieval effectiveness. https://www.semanticscholar.org/paper/Search-strategy-and-the-optimization-of-retrieval-Salton/a4b1ebd5e7a3fca836b4b037ae78ff176620ef51.

Salton, G., Wong, A., Yang, C.S., 1975. A vector space model for automatic indexing. Commun. ACM 18 (11), 613–620. http://dx.doi.org/10.1145/361219.361220.

Sanderson, P.C.M., 2013. Evaluating the performance of information retrieval systems using test collections [Text]. Professor T.D. Wilson. https://informationr.net/ir/18-2/paper582.html#Xs9hBsBRXIU.

Shadiev, R., Hwang, W.-Y., Chen, N.-S., Huang, Y.-M., 2014. Review of speech-to-text recognition technology for enhancing learning. Educat. Technol. Soc. 17, 65–84.

Singh, J.N., 2012. Analysis of vector space model in information retrieval.

Singh, D., Adeku Musa, I., Tella, Y., Singh, J., 2007. An overview of the applications of multisets. Novi Sad J. Math. 37, 73–92.

Tsatsaronis, G., Panagiotopoulou, V., 2009. A generalized vector space model for text retrieval based on semantic relatedness. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop on - EACL '09. pp. 70–78. http://dx.doi.org/10.3115/1609179.1609188.

Vollmer, D., 2015. Natural language processing of morphology with linguistically motivated applications to German linking elements.

Wahyudi, E., Sfenrianto, S., Hakim, M.J., Subandi, R., Sulaeman, O.R., Setiyawan, R., 2019. Information retrieval system for searching JSON files with vector space model method. In: 2019 International Conference of Artificial Intelligence and Information Technology. ICAIIT, pp. 260–265. http://dx.doi.org/10.1109/ICAIIT.2019.8834457.

Wang, et al., 2018—GLUE A Multi-Task Benchmark and Analysis Platform.pdf. (n.d.).

Wintana, D., Sfenrianto, Hikmatulloh, Raharjo, M., Putra, J.L., Ambarsari, D.A., Jayanti, D.D., 2019. Searching Information Tourism using Vector Space Model: Proceedings of the 2nd International Conference on Applied Science, Engineering and Social Sciences. pp. 247–251. http://dx.doi.org/10.5220/0009882102470251.