A chunking-based text-tokenization framework incorporating domain knowledge

Anonymous ACL submission

Abstract

One of the crucial activities in Natural Language Processing (NLP) is to tokenize text to extract features so that data mining models can be applied. Many widely-used tokenization algorithms take the approach of using words as tokens. This approach suffers from the following limitations: (a) Using words as features leads to high dimensionality of the data file generated from text, (b) These algorithms use a one-size fits all approach on the text and extract tokens uniformly without consideration of the prior knowledge available in the domain. Here a novel method is proposed which extracts features by tokenizing text as chunks. Domain specific knowledge is used to generate syntactic rules which are then used to split text documents into Finite State Rule Based Chunks. The chunks are the tokens on which data mining models are then applied to generate insights. The effectiveness of chunk-based tokenization is demonstrated by extracting chunk-based tokens as well as word-based tokens from a document corpus, and performing clustering in both cases. A comparison of the clustering of corpus with chunk tokens vis-à-vis word tokens shows marked improvement in clustering performance in the former.

1 Introduction

With the advent of machine learning and natural language processing (NLP) tools, huge volumes of text data can be mined and patterns, sentiments, latent topics identified with reduced human effort. One of the major problems in NLP is generating features from text. Some of the popular methods involve embedding either the individual words in the text (also called tokens) or a combination of two, three, or n-words (also called bigrams, trigrams, n-grams, etc). The limitations in these approaches are threefold: a) Embedding individual words as tokens leads to generation of high-dimensional vectors for each document which leads to creation of sparse matrices and overfitting risk; b) While bi-grams, tri-grams or n-grams help in reduction of dimensions of the embedding vectors, every feature thus generated consists of two, three or n-words respectively as a one-size-fits-all solution, without considering that different semantic meanings might require different customizations; c) The involvement of prior domain knowledge in the traditional token embedding or n-gram embeddings is minimal.

In this paper, a text featurization framework is proposed that uses user's prior domain knowledge to extract specific customized phrases from the text as features, also called as chunks. Using domain specific attributes, various syntactic rules are developed which are then used for splitting text documents into chunks. This method is called Finite State Rule Based Chunking, where the syntactic rules generated are used to identify chunks from text documents using tag patterns (sequence of part of speech tags). Once the chunks are identified, clustering is applied to group the chunks based on their semantic similarity, thus reducing the chunks to a standardized list of cluster labels. Using the cluster labels as features, the documents are then embedded. This method helps avoid the high dimensionality and sparsity of the other embedding frameworks and provides better performance.

The remainder of this paper covers the methodology and presents a case study showcasing the application of this model. Section 2 covers the gaps and challenges in application of NLP models and techniques. Section 3 describes the concept of chunking and the broad methodology. A

demonstration of model application is given in a case study in section 4 where safety incident investigation reports generated in a steel plant are analyzed to identify chain of events model of accident causation. TFIDF vector space model is used for converting the chunks into feature vectors and perform clustering. The findings and comparison of the chunking results are also presented. Section 5 provides a comparative analysis of clustering performance of chunking vis-à-vis original documents. Finally, sections 6 and 7 provide the conclusions, limitations and future scope of work.

2 Literature Review

Text mining refers to knowledge discovery by identifying high-relevance patterns from text (Feldman and Dagan, 1995). Allahyari et. al. (2017) classified various text mining approaches as Information Retrieval (IR), text summarization, Natural Language Processing (NLP), sentiment analysis, opinion mining and Information Extraction (IE). Zhang et. al. (2015) classified text mining algorithms into four classes: text categorization, trend modelling, text clustering and association rule mining. Text mining uses both supervised and unsupervised models. Some of the common models used in supervised models include nearest neighbor classifiers, decision trees, rule-based classifiers and probabilistic classifiers (Sebastini, 2002). Some popular categorization techniques in supervised models include KNN, SVM, Naïve Bayes etc (Kadhim, 2019).

While supervised algorithms offer advantages in terms of obtaining precise, user-friendly insights and labels from historical data (in form of labelled training data), in big datasets, there is requirement of a large annotated corpus and it becomes expensive and time consuming to do annotations. Secondly, when there is some modification in the domain, it requires re-annotation of the training data all over again. Hence unsupervised algorithms become potentially important. Use of probabilistic techniques for text mining include topic modelling using probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999) and Latent Dirichlet Allocation (LDA) (Blei, 2012). The objective of topic modelling is to retrieve suitable latent topics from document collections. For example, Chang et al (2009) used quantitative methods for evaluating semantic meaning in topics inferred from probabilistic topic modelling. While topic models are useful for identifying the prevalent themes in a document saving time and effort, owing to their generalization they face many limitations: (a) They do not provide insights on hierarchical relationships (b) They are difficult to be expressed in a meaningful way. Hence some variations of topic models have been proposed e.g. supervised topic models for classification and regression respectively accounting for annotators' heterogeneity and bias (Rodrigues et. al., 2017), discriminative relational topic models (Chen et. al., 2014) which is a generative process that combines modeling of the document link structure with document contents, and differential topic models (Chen et al, 2014]) that use hierarchical Bayesian nonparametric techniques to model both topic differences and similarities.

Another major challenge faced by both supervised and unsupervised models is in text representation. Currently the most widely used representation is the vector space model (VSM) (Stavrianou et. al., 2007) where a string of text is described by a vector of text features and its content contains a function of the feature attributes (e.g. frequencies with which these features appear in the corpus or corpora, relevance of the features etc) (Salton et. al., 1975). Various kinds of text representation models are used which are extensions of the VSM. Some representations focus on phrases instead of single words (Blake and Pratt, 2001; Caropreso et. al, 2001; Mitra et. al., 1997) and some give importance to semantics of words or relations between them (Cimiano et. al., 2005; Kehagias et. al., 2003; Rajman and Besancon, 1999) while some take advantage of the hierarchial structure of the text (Antonellis and Gallopoulos, 2006). These models either use term by sentence matrix (Antonellis and Gallopoulos, 2006), association rules (Blake and Pratt, 2001), combination of bag of words and concept hierarchy (Bloehdorn et. al., 2006), n-grams (Caropreso et. al., 2001), concept hierarchy (Cimiano et. al., 2005), supervised term weighing method (TF-RF) (Lan et. al., 2008), Latent Semantic Indexing where the semantic structure of the documents is used to improve detection of relevant documents on the basis of query terms or sense-based vectors (Kehagias et. al., 2003). Thus, while these models address some of the problems faced during text representation, they lack in some other areas e.g. (i)

Lack of incorporation of prior knowledge during text representation (ii) fixed n-gram models (iii) word hierarchy is unaddressed in some of the models.

By choosing Finite-State Rule Based Chunking, we propose certain advantages: (i) Use of prior domain knowledge in application of unsupervised models will generate insights that will be relatable to the domain user and allow for better interpretation (ii) rule-based chunks have flexibility of size. Instead of rigidly defining n in n-grams, rule-based chunks can be unigrams, bigrams trigrams and so on (iii) many text documents cannot be classified into a single class rather contain subsections that can be classified into different classes. Chunking based classification enables splitting a document into various subsections and separately classifying the subsections into different classes (iv) rule-based chunks can incorporate word associations and represent the hierarchy of insights. (v) this prevents high-dimensionality of the embedding vectors for each document, this reducing the risk of overfitting.

Once the text is split into chunks, the model proposes application of various NLP models on them for pattern recognition and knowledge discovery. In the case study presented here, in analysis of incident investigation reports, to identify the events that form a chain that leads to an accident, the reports are split into various chunks where each chunk is representative of an event. Then K-means clustering is applied on the chunks to identify the major attributes. A comparison is made between clustering of original text documents vs clustering of text chunks, and it is observed that clustering of chunks, instead of whole documents, provides better performance.

3 Methodology

3.1 Theory of chunking

Chunking is the process of extracting non-overlapping segments from a document. These segments are the basic non-recursive phrases and named entities that correspond to major parts-of-speech. Hence a chunk is a set of tokens that form a syntactically correlated phrase.

A chunker is developed in two ways: Finite-State Rule Based Chunking and statistical approach based chunking. In statistical approach to chunking, a training dataset is annotated with POS tags and chunk class tags and the chunker model is trained on it. On the contrary, in Finite-State Rule Based chunking, a set of hand-crafted syntactic rules is used to identify the prior knowledge based themes in the document and capture the relevant phrases from the text. Hence unlike statistical chunking that requires annotated data, finite element chunking requires POS tags of tokens. These hand-crafted rules define the chunk grammar structure which becomes the framework for chunk phrase extraction. E.g. a noun phrase chunk (NP chunk) will consist of a noun token, adjective token etc. whereas a verb token will consist of verb token, adverb token and noun token.

In this study, since domain knowledge is being combined with unsupervised models, an ensemble framework of Finite-State Rule Based chunker combined with clustering is proposed.

3.2 Methodology

The first step of the methodology involves understanding the problem statement. This enables rule crafting as per the objectives and domain expertise and then use the rule for extraction of phrases from text using Finite-State Rule Based Chunking. From the rules, tag patterns are identified and the POS tags of words are then used to identify the chunk phrases of words that conform to the tag patterns. NLTK Python library is used for this purpose. Hence the chunks encapsulate prior domain knowledge. These chunks then become the new tokens/features for text mining models to apply on.

The chunks are then converted into feature vectors. As the chunk phrases extracted will contain differences owing to presence of different words or morphological variants of the same word, it is necessary to group semantically similar chunks and for that clustering is utilized. For that purpose, we postulate that each chunk/phrase can be considered as separate document representing a building block of the semantic structure of the original document. A collection of such chunks can be considered as a new corpus. The words in the chunk corpus are then TF-IDF vectorized to form a global TF-IDF matrix. K-Means clustering is applied on the TF-IDF vectors and the chunks are classified into clusters. The cluster word-clouds are then analyzed to identify primary attributes for

each cluster and give an appropriate cluster label. The label is then applied to annotate all the chunks belonging to that cluster. Once the chunks are annotated with the parent cluster labels, the chunks are re-assembled to form the original document and the chunks are then substituted with the respective cluster labels. These cluster labels then become the new tokens of the document. Then considering that the document is made of these new cluster label tokens, the tokens are TF-IDF vectorized and K-Means clustering is carried out to cluster the original documents.

4 Case Study: Identification of chain of events at a steel plant

To demonstrate chunk effectiveness, we have considered a corpus of accident reports that happened at a steel plant from 2015-2018. 636 incident descriptions were collected. These described the root cause behind accidents and what were the sequence of unsafe events/activities that escalated to the accident. The objective is to cluster the documents by using the properties of the unsafe events/activities.

4.1 Finite-State Rule-Based Chunking

After pre-processing of the text data, the POS tags of words are noted. Identifying the POS tag of each word is crucial to do chunking which will use these POS tags to identify the tag patterns of the various chunks.

An accident is caused by escalation of a sequence of events that lead a Hazardous Element to cause harm. These activities an actor and an action. The action is represented by a verb phrase and the actor is represented by a noun phrase. So there are two parts to this: A noun phrase (represented by adjective-noun combination) that cause the verb phrase (verb-adverb combination) to occur.

For Example, if the following incident was recorded "Toxic gas leaked without being detected" the verb-adverb combination comprising the Verb Phrase "leaked without being detected" represents the action that can escalate to an accident. The actor is represented by "Toxic gas" which is an adjective-noun combination forming the Noun Phrase (NP). The Verb Phrase was caused by the Noun Phrase.

Hence the tag pattern to be developed for the unsafe event is as follows:

<adjective><noun><verb><noun/adverb>

Using this tag pattern all such chunks/phrases representing the unsafe event can be extracted.

4.2 K-Means clustering using TF-IDF vector

In this method, each chunk is considered as a separate document and corpus from all these chunks are formed. These chunk corpus is then TF-IDF vectorized to obtain the TF-IDF matrix of all the unique words in the corpus. In the steel plant accident report corpus, 1708 chunks were identified. Listing the 1708 chunks into a corpus, 1923 unique tokens were obtained. Hence the vectorization leads to a 1923 x 1 vector generated for each chunk. The global corpus embedding matrix is thus of size 1708 x 1923. Using elbow method, the optimal number of clusters obtained is 28, and K-means clustering is performed. As the clusters are formed in a 1923-dimensional space, to plot them on a 2D space, t-SNE algorithm is used. The clusters are shown in Fig 1. Word-clouds are obtained for the clusters to identify the keywords that identify the major attributes of the clusters. The word-clouds are shown in Fig 2.

These attributes will be used to annotate the 28 clusters with appropriate cluster labels which will automatically annotate all the chunks present in that cluster and derive the information present in various chunks. The cluster labels are given in Table 1.

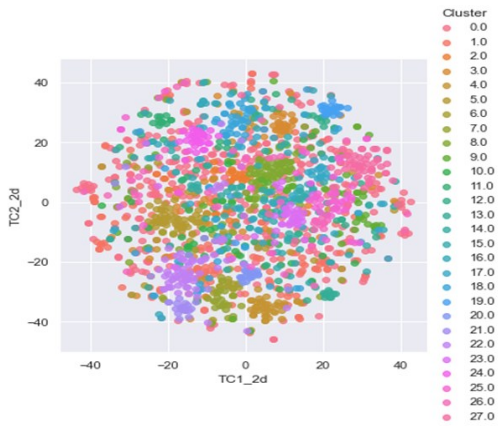

Figure 1: Distribution of clusters

Figure 2: Cluster word-clouds

Table 1: Annotation of the clusters based on word cloud keywords

Clusters	Major attributes
Cluster0	Equipment malfunction
Cluster1	Slide gate operations issue
Cluster2	Crane, ladle and car and related operations related issues
Cluster3	Metal, oil and gas leakage
Cluster4	Falling of liquid metal
Cluster5	Hazardous water related incidents
Cluster6	Gas leakage, exposure, ignition related incidents
Cluster7	Cast and tap hole operations issues
Cluster8	Related to personnel activities and workers' exposure to various safety issues
Cluster9	Fire related issues
Cluster10	Miscellaneous activities (cluster can be ignored in analysis)
Cluster11	Slag and slag equipment related issues
Cluster12	Furnace related issues
Cluster13	Equipment operation issues
Cluster14	Pipe, line and pipe equipment related problem
Cluster15	Flame and smoke related issues
Cluster16	Administering of first aid
Cluster17	Tanks and chemicals related issues
Cluster18	Emergency situations created w.r.t equipment, personnel, hazardous substances
Cluster19	Hot metal/molten metal related issues
Cluster20	Carbon monoxide gas leakage/exposure
Cluster21	Miscellaneous activities of equipment (cluster can be ignored)
Cluster22	Vessel related issues
Cluster23	Valve related issues
Cluster24	Various kinds of incidents happening
Cluster25	Issues due to high/low pressure
Cluster26	Battery and explosion related issues
Cluster27	Issues due to air equipment, air pressure etc

A sample case is demonstrated here where the cluster labels for the chunks are used to define the new tokens. The original document and chunks identified are given in Table 2. The chunk trees are developed as demonstrated in Fig 3.

Table 2: Example of chunk tree formation

Incident Description	ACM chunks
L1 level indicator shown 5 %, all dept1 pumps tripped and diesel pump started	['level indicator shown pump', 'diesel pump started']

After identifying the chunks in the document the chunks are substituted with respective cluster label to obtain the new standardized text document, as shown in Table 3:

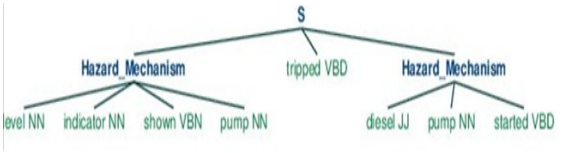

Figure 3: Chunk tree for accident causing mechanisn

Table 3: Clusters of chunks using K-means clustering

Chunk	Cluster
level indicator shown pump	0
diesel pump started	13

Hence the corpus of original text will look as shown in Table 4:

Table 4: Documents with cluster labels as tokens

Description
C6 C8 C0 C26 C24
C0 C13
C25 C23
C6 C26 C0 C23 C23 C23 C23 C0
C26 C11 C9
C19

So now each document in the corpus is considered to consist of cluster labels only as the new words. These labels are then tokenized and embedded using TF-IDF to generate embedding matrices of size 636 x 28, where 636 is the number of documents, 28 is the number of cluster labels representing the tokens and for each document, the IF-IDF gives the term frequency-inverse document frequency of each cluster label. On this embedding matrix, K-means clustering is carried out. This clustering is based on chunk as features which incorporate information about unsafe events, hence documents showing similarity in unsafe events will get clustered together.

5 Results and Discussion

First, to analyze if splitting a text document into chunks provides any advantages, clustering of the corpus obtained by listing of all chunks is compared with clustering of the corpus of original documents. Two cluster validity indices- Davies-Bouldin index (DBI) and Silhouette Index (SI)- are used to evaluate the clustering models. The results are showcased in Table 5:

Table 5: DBI and SI values of clustering models

	TFIDF based clustering of chunks	TFIDF based clustering of original documents
DBI	4.596	5.405
SI	0.031	0.017

The following observations can be made:

a) A lower DBI score indicates better clustering. Hence it can be observed that chunking + clustering is giving better clusters compared to clustering original documents

b) If the SI value is closer to 1, the clustering is better. Higher value of SI for clustering of chunks suggests that chunking and clustering is better than clustering of original documents

Then clustering performance of original documents based on the 636 x 28 embedding matrix is compared with clustering of TF-IDF embedding of the original documents. The SI and DB index of both the clustering is shown in Table 6:

Table 6: Clustering performance of chunk-documents vs Original Documents

	Clustering of documents using the cluster-label embedding matrix	TFIDF based clustering of original documents
DBI	2.7885	5.405
SI	0.1208	0.017

Hence it can be seen that clustering based on chunks is giving better results.

6 Conclusion

Traditional NLP models classify a text document into various classes. However, at times many text documents have a structure where various portions of the document belong to various classes and it is not feasible to classify an entire text document into a class. While identifying chain of events,

identification of these subclasses is important. Hence this framework has been proposed which enables classification of various subsections into various classes and enables development of chain of events. Demonstration of the framework is conducted in form of a case study where the study was conducted to find out how to do text mining of safety text data generated by an organization and generate chain of events/accident paths with minimal human involvement. In the case study, unlike other NLP models which generate numerical values of tokens based on a global corpus, here the rules obtained from hazard theory have been used to guide the formation of initial dataset which is then analyzed. On changing domains, the rules also change, hence this model is useful to obtain domain specific insights.

By adopting an ensemble approach of domain specific lexical rule-based chunking with unsupervised NLP tools, this model aims to bridge the drawbacks of both supervised and unsupervised approaches:

One major drawback of supervised modelling is annotation of huge corpus of training documents. By splitting a document into chunks and performing clustering on them, all similar chunks, representing a particular concept, get annotated at once just by annotating the clusters. Secondly, text documents can have multiple annotations for various sub-sections which might not get captured during manual annotation. Splitting the documents into chunks and annotating using clustering allows for a text document to have multiple annotations for different subsections.

Unsupervised models are domain independent, hence combining these models with chunking enables the user to incorporate user's prior knowledge of domain features into generation of insights. This helps in domain relevant interpretation of insights and thus resolve the difficulty of interpretation of insights obtained from unsupervised models.

While unsupervised models are useful to model latent topics and themes, one of the key issues is difficulty of modelling hierarchy and relationship between themes. Chunking of a document organizes topics and themes in a hierarchial manner, which will then be discovered using NLP models. Chunks and the relationships they represent help in customized information discovery from the document.

Finally, in the case study, the original document corpus embedding using TF-IDF generated 1708 x 1923 embedding matrix, while using chunking + clustering and then doing TF-IDF embedding generates 1708 x 28 embedding matrix. This reduces the risk of high dimensionality and overfitting.

7 Limitations and Future Work

Despite the advantages, the model suffers from a few drawbacks. Removal of stop words and connecting words also helps in chunks becoming less user friendly to understand. Secondly, the model makes an assumption that the descriptions are written in a sequence, and the sequence is used to obtain the chain of events. If the descriptions are written in a haphazard manner, then this model will not give optimal results. Thirdly, using K-Means clustering does not always capture the semantic nature, even in chunks.

Keeping in mind the limitations, the following future research directions are proposed:

a) Future work will involve how to remove these drawbacks while doing chunking analysis.
b) Future work will include how to better cluster chunks capturing the semantic nature.
c) Application of other feature vector generation models on chunking to improve insights generated.

References

Ronen Feldman, and Ido Dagan. 1995. Knowledge Discovery in Textual Databases (KDT). In *KDD* (Vol. 95, pp. 112-117).

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. 2017. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*.

Yu Zhang, Mengdong Chen, and Lianzhong Liu. 2015. A review on text mining. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 681-685. IEEE.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 34, no. 1: 1-47.

Ammar Ismael Kadhim. 2019. Survey on supervised machine learning techniques for automatic text classification. *Artificial intelligence review* 52, no. 1: 273-292.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*.

David M. Blei. 2012. Probabilistic topic models. *Communications of the ACM*, *55*(4), pp.77-84.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems,* 22.

Filipe Rodrigues, Mariana Lourenco, Bernardete Ribeiro, and Francisco C. Pereira. 2017. Learning supervised topic models for classification and regression from crowds. *IEEE transactions on pattern analysis and machine intelligence* 39, no. 12: 2409-2422.

Ning Chen, Jun Zhu, Fei Xia, and Bo Zhang. 2014. Discriminative relational topic models. *IEEE transactions on pattern analysis and machine intelligence* 37, no. 5: 973-986.

Changyou Chen, Wray Buntine, Nan Ding, Lexing Xie, and Lan Du. 2014. Differential topic models. *IEEE transactions on pattern analysis and machine intelligence* 37, no. 2: 230-242.

Anna Stavrianou, Periklis Andritsos, and Nicolas Nicoloyannis. 2007. Overview and semantic issues of text mining. *ACM Sigmod Record* 36, no. 3: 23-34.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18, no. 11: 613-620.

Catherine Blake, and Wanda Pratt. 2001. Better rules, fewer features: a semantic approach to selecting features from text. In *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 59-66. IEEE.

Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. 2001. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. *Text databases and document management: Theory and practice* 5478, no. 4: 78-102.

Mandar Mitra, Chris Buckley, Amit Singhal, and Claire Cardie. 1997. An analysis of statistical and syntactic phrases. In *RIAO*, vol. 97, pp. 200-214.

Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of artificial intelligence research* 24: 305-339.

Athanasios Kehagias, Vassilios Petridis, Vassilis G. Kaburlasos, and Pavlina Fragkou. 2003. A comparison of word-and sense-based text categorization using several classification algorithms. *Journal of Intelligent Information Systems* 21: 227-247.

Martin Rajman, and Romaric Besançon. 1999. Stochastic distributional models for textual information retrieval. In *Proc. of 9th International Symposium on Applied Stochastic Models and Data Analysis (ASMDA-99)*, pp. 80-85.

Ioannis Antonellis, and Efstratios Gallopoulos. 2006. Exploring term-document matrices from matrix models in text mining. *arXiv preprint cs/0602076*.

Stephan Bloehdorn, Philipp Cimiano, and Andreas Hotho. 2006. Learning ontologies to improve text clustering and classification. In *From Data and Information Analysis to Knowledge Engineering: Proceedings of the 29th Annual Conference of the Gesellschaft für Klassifikation eV University of Magdeburg, March 9–11, 2005*, pp. 334-341. Berlin, Heidelberg: Springer Berlin Heidelberg.

Man Lan, Chew Lim Tan, Jian Su, and Yue Lu. 2008. Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence* 31, no. 4: 721-735.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, no. 6: 391-407.