Supporting Trainset Annotation for Text Classification of Incoming Enterprise Documents

Juris Rats[a] and Inguna Pede
RIX Technologies, Blaumana 5a-3, Riga, LV-1011, Latvia

Keywords: Machine Learning, Trainset Annotation, Text Clustering, Text Classification, More like This Query, Elasticsearch.

Abstract: Volumes of documents organisations receive on a daily basis increase constantly which makes organizations hire more people to index and route them properly. A machine learning based model aimed at automation of the indexing of the incoming documents is proposed in this article. The overall automation process is described and two methods for support of trainset annotation are analysed and compared. Experts are supported during the annotation process by grouping the stream of documents into clusters of similar documents. It is expected that this may improve both the process of topic selection and that of document annotation. Grouping of the document stream is performed firstly via clustering of documents and selecting the next document from the same cluster and secondly searching the next document via Elasticsearch More Like This (MLT) query. Results of the experiments show that MLT query outperforms the clustering.

1 INTRODUCTION

Both supervised and unsupervised machine learning methods are used for classification tasks. The disadvantage of supervised learning is they rely on annotated data sets that take considerable resources to prepare. Supervised learning is still the preferred approach for dealing with a major part of classification tasks. This is a case in particular for text classification. Unsupervised methods are found to perform not very well for domains featuring high dimensionality data representations as text classification tasks do. This rules out unsupervised learning methods as the main method for text classification.

Unsupervised learning still can be used as a support method to reduce human involvement in the labour-intensive task of trainset preparation – namely sample annotation.

The annotation of trainset for text classification comprises two intertwined processes – the creation of topic set and labelling of texts with topics. We assume that this process can be improved by organizing the stream of text samples in groups of similar texts. This would help both to create the appropriate topic set and to reduce the manpower necessary for sample annotation.

We research and compare here two technologies in this respect:

- clustering methods;
- Elasticsearch More Like This query (MLT).

The result in either case will be a method that improves the sequence of samples. In the case of clustering the next sample will be fed to the expert from the same cluster (if there are any). In case of MLT – the next sample will be the most similar found by the search. We analyse thus a number of clustering workflows as well as MLT configurations. The solution is considered to be better if the overall probability of feeding the sample of the same topic is higher.

Clustering pipeline usually comprises several steps of machine learning domain – data cleaning, feature embedding, feature reduction and the clustering. Chapter 2 outlines the related work in these domains as well as application of MLT to similar tasks.

[a] https://orcid.org/0000-0002-3406-7540

Rats, J. and Pede, I.
Supporting Trainset Annotation for Text Classification of Incoming Enterprise Documents.
DOI: 10.5220/0011113000003269
In *Proceedings of the 11th International Conference on Data Science, Technology and Applications (DATA 2022)*, pages 211-218
ISBN: 978-989-758-583-8; ISSN: 2184-285X

2 RELATED WORK

Machine learning methods have been used for text classification in a number of domains as sentiment analysis (Avinash M & Sivasankar E, 2019; M. Fu et al., 2018; Maas et al., 2011; Pang & Lee, 2008), news classification (Kadriu et al., 2019), web page classification (Shawon et al., 2018) etc. Text classification mainly is based on supervised machine learning methods spanning a number of basic methods like Naïve Bayes classifier, K-Nearest Neighbours, Support Vector Machines as well as deep learning models (Kowsari et al., 2019).

Text classification solution must address two important areas – embedding and dimension reduction.

A number of embedding methods are used to get convenient text representation.

One of the core models used for text embedding is Bag of Words (BoW) (Harris, 1954). BoW has been applied in number of domains, e.g. paraphrase generation (Y. Fu et al., 2020), biomedical concept extraction (Dinh & Tamine, 2012) and recommender systems (Bayyapu & Dolog, 2010). One of the best known implementations of the BoW method is tfidf (Joachims, 1997). Bag of meta-words (BoMW) is supposed to improve the original BoW method using meta-words for embedding (M. Fu et al., 2018).

A number of alternative models have been developed that use context (surrounding words) of a word to create word embedding. This allows to create embeddings putting words with a similar meaning close to one another in a representation space. The assumption here is that words that have similar context have to have similar meaning. Examples of such embedding methods are GloVe (Pennington et al., 2014), word2vec (Mikolov et al., 2013), doc2vec (Le & Mikolov, 2014) and others. These are followed by so called contextual models BERT, ELMo, GPT, XLNet (Liu et al., 2020; Ular & Robnik-Šikonja, 2020), Sentence-BERT (Cygan, 2021) and others.

Advanced embedding methods like BERT, GPT and XLNet have demonstrated good results for a number of tasks still they are resource hungry and thus must be applied only in domains where they are considerably better than basic methods like tfidf.

Text embedding is a vector of high dimensionality as a rule. Some kind of dimensionality reduction techniques is normally used to save resources and/or improve the performance of text classification or other downstream tasks. The methods used include PCA (Principal Components Analysis) (Taloba et al., 2018), truncatedSVD (Hansen, 1987) and UMAP (Uniform Manifold Approximation and Projection) (McInnes et al., 2018). Dimensionality reduction is vital both for supervised and unsupervised learning.

Supervised learning is superior for text classification tasks. The better performance of supervised learning algorithms is based on knowledge accumulated in annotated trainsets. The disadvantage here is that one may invest considerable resources to create those annotated trainsets. It is therefore of great interest to develop methods that could reduce the manpower necessary for trainset annotation.

A number of topic modelling methods have been used to identify latent topics in unlabelled text corpora. LDA (Latent Dirichlet Allocation) has emerged as the mainstream method lately (Balakrishnama & Ganapathiraju, 1998; Tong & Zhang, 2016) and domains explored include news articles, Wikipedia articles, product and customer reviews, software change requests etc. The main challenge with the topic modelling is how to match a topic model automatically generated from the text corpora to the "real" topics of the domain (Jacobi et al., 2016). Experiments with LDA on documents of our domain of interest (stream of incoming documents of the organization) showed that the topics models generated do not match well enough to use this method for the identification of the meaningful topic set and for text labelling.

The clustering methods generally can be split into two groups – ones that need to know the resulting cluster count in advance and ones that don't. We are interested in clustering methods that don't need a cluster count to be pre-set. Namely – we use in our research hdbscan (Hierarchical Density-based Spatial Clustering of Applications with Noise - (Malzer & Baum, 2019)) and birch (Balanced Iterative Reducing and Clustering Using Hierarchies - (Zhang et al., 1996)).

Unfortunately the clustering methods do not perform well for high dimensionality data (Assent & Seidl, 2009; Parsons et al., 2004; Steinbach et al., 2003). Text clusterization features both high dimensionality and large numbers of clusters (Karpov & Goroslavskiy, 2012). Still clustering could be used to support annotation of the training data of supervised method.

The mainstream approach to text classification is the supervised machine learning. Still there are alternatives. One of them we came across is the contextual MLT query provided by Elasticsearch (Klinger, 2019). This approach has been applied for phrase classification (Yellai, 2016) recommendation engine and duplicate detection (Vola, 2017). MLT is

used to search documents similar to given set of source documents / texts (see Chapter 6).

3 DOMAIN DESCRIPTION

A stream of incoming documents inundates organizations and businesses at an ever-increasing speed and this forces organizations to invest more resources to index this document flow (to make documents searchable and to route them inside organisation) promptly and correctly. Therefore, automation of the document indexing is an important and urgent task.

Enterprise documents differ from the types of text generally used for topic modelling or classification tasks (like news articles). Guided by observations in several organizations we assume that:

- documents may be lengthy and have various layouts with each part having a different role;
- documents belong to a large number of topics, still the document set is highly unbalanced – a handful of topics cover most of the document amount;
- document topics and distribution of documents between topics may change over time;

Unbalanced nature of the document set (see sample distribution on Figure 1) means, in particular, that most of the topics have a rather small number of documents belonging to them. This makes them inconvenient for automation because of a lack of training data. We propose thus to focus on automation of handling of the largest topics and leave the rest to manual handling as before.

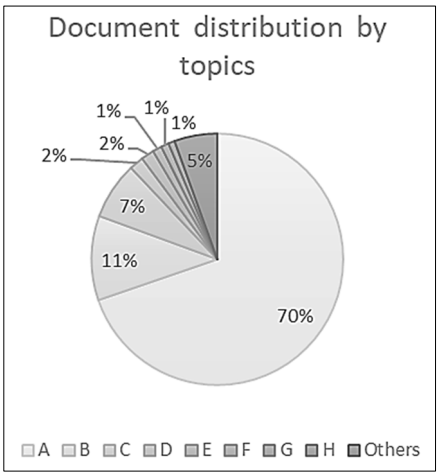

Figure 1: Document distribution by topics.

4 HANDLING INCOMING DOCUMENT AUTOMATION

A document generally has some attached metadata (like sender and addressee). Our model uses this metadata in concert with document topics to create document clusters for valid indexing (i.e. a document cluster unequivocally determines indexing rules to be applied).

Assuming the document metadata are available (or can be retrieved) upon the document reception we focus in our research on automation of the process of assigning topics (labels) to text data. Sets of annotated training samples (samples with topics assigned) must be created in advance to use supervised learning methods for the classification. Experts face a number of challenges when creating the annotated trainset:

- a convenient set of topics must be created;
- a sufficient amount (to train classification method) of samples must be provided with correct labels

We assume that it is possible to create a set of topics such that a topic (possibly in combination with document sender and/or receiver data) unequivocally determines the indexing of the document. Document sender/receiver may be identified by other means (out of scope of this research) therefore as long as the topic of the document is identified, the handling of the incoming document is determined.

We propose to create a set of topics and annotate historical documents beforehand (setup process). A sufficient number of documents have to be annotated to understand what are the main topics and to create initial trainsets for them (we annotated about 1000 documents). When ready with the setup process we switch to the main process where new incoming documents are handled by the model. Some details of both processes are below.

4.1 Setup

Setup process involves experts to annotate a set of historical documents of the organization. The main goal here is to create topics that:

- unequivocally determine the handling of its documents;
- are separated well enough one from another.

The first depends fully on expert's domain knowledge. We aim to support the second by manipulating the order of documents presented to experts (only the order is influenced, all the documents of the set are processed anyway). Namely

– we would like to increase the likelihood that the next document in the stream of documents presented to expert has the same topic as the current one. This should help to create longer sequences of the same topic documents and thus help both to create a topic set and to annotate the documents. Having in mind the unbalanced nature of the document streams the aim of the setup process is to identify the largest (major) topics while all other topics are merged into one "others" topic.

Two approaches for determining the next document (that would increase the likelihood it will have the same topic) are explored and compared in this research:

- clustering the documents in advance and selecting the next document from the same cluster as the current one (see Chapter 5);
- using MLT to find the next document (see Chapter 6).

The setup process should result in creation of a set of the largest topics (we call them major topics) and sets of annotated trainsets for them.

4.2 Main Process

Our model provides means for creating and maintaining a domain specific indexing rule set. Each rule here defines indexing for a particular combination of topic and document metadata (e.g. document sender). Metadata values are retrieved from the document automatically while topic may be assigned either manually by an expert or automatically using classification bots. Setting of the document topic thus automatically triggers an indexing rule.

The main process (Figure 2) is executed on a set of topics created during the setup and uses text classification bots (see explanation in this section below) trained for the largest topics. A bot is trained as long as number of documents of the respective topic reaches the (configurable) threshold and is used for prediction if the bots validated performance reaches the configured score. The performance (prediction precision and recall) levels for automatic and suggestion modes may be configured. Setting (manually or automatically) the topic of the document invokes the linked indexing rule.

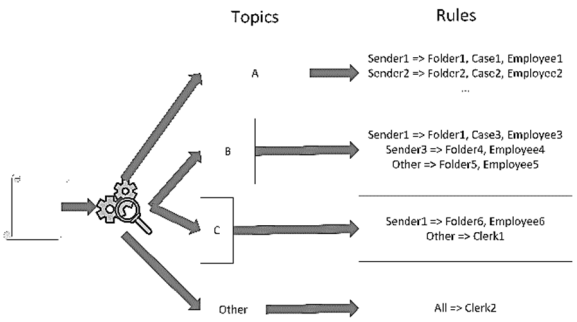

Figure 2: The main process.

A bot for a major topic is trained to predict if a given document belongs to the respective topic. The bot is trained on a balanced trainset consisting of an equal share of positive (documents of a bot's topic) and negative (documents of other topics) samples.

Important issue to keep in mind here is that a penalty of false positives is higher in our case than that of the false negatives. False negative means in our case that a document is not assigned a particular topic by a classification bot. The case is handled then by a clerk who will assign (hopefully) the correct topic. In case of a false positive a wrong topic might be assigned and the document might be indexed and routed not as expected. This means negative experience for the person receiving incorrectly routed document.

To mitigate the effect of false positives our model does not finalize the topic assignment until the employee the document is routed to accepts the topic assignment. In case the employee rejects the topic assignment the document is routed back to a clerk who can assign the valid topic.

5 TEXT CLUSTERING

As pointed out above (see Chapter 4.1) we use text clustering as one of the technologies for the first step of the process. Our assumption (supported by the experiments) was that the clusters will match at least partly the document topics. This should help experts to better understand what the major topics are, to increase the quality of trainsets and to reduce the time necessary for document annotation.

One of the challenges here was to understand what combination and configuration of embedding, feature reduction and clustering methods are the best for the document stream in question. We performed a set of measurements on clusters for the document set after it was annotated by experts. Each of the cluster sets created by a particular combination of methods and hyperparameters:

- was evaluated using a-priori metrics (Silhouette, Calinski-Harabasz and Davies-Bouldin indexes);

- was compared using Rand index (Warrens & van der Hoef, 2020) to the document distribution according to the topics.

Rand index measures how well clusters match the topics. The problem is we cannot calculate the Rand index in advance therefore we cannot use it to decide on the best clustering combination. What we can do is to use Rand index evaluations of the clustering configurations when implementing the model for the next document streams. Another problem with Rand index for our case is discussed below in section 5.2.

5.1 Comparing Clustering Metrics

Figure 3 reveals the results of one of the performance evaluations for combinations of embedding (tfidf, doc2vec, bert), feature reduction (truncatedSVD and umap) and clustering (hdbscan and birch) methods. Rand index for documents of largest (2 to 6) topics are shown as well as values of silhouette index. Values of Rand indexes are shown on the leftmost y-axis, silhouette index – on the rightmost. Configuration combinations are ordered by values of RAND for 2 largest topics. The line diagram is used here just to show trends, x-axis has discrete values.

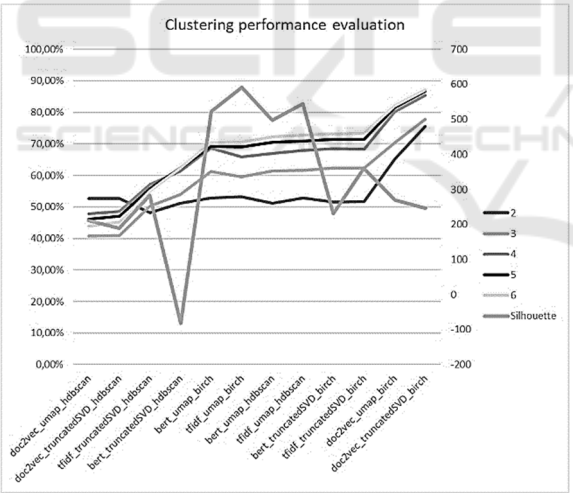

Figure 3: Clustering performance.

We can see here that umap reduction method in combination with bert or tfidf embedding has the highest silhouette values. Rand index for the said combinations are good, but the highest here are other two combinations of doc2vec embedding with birch clustering. Other a-priori metrics show similar results – the best values correspond to good values of rand index, but did not identify the combinations with the best rand index. This means that a-priori evaluation methods in case of no additional information may be used to select a good clustering configuration.

5.2 Similarity Score

Rand index penalizes symmetrically both for items of a cluster having different topics and for topic items in different clusters. The first is important in our case as this indicates that clusters have documents of mixed topics and this decreases the likelihood of the next document (taken from the cluster) to have the same topic. The second means the documents of the topic are split into several clusters. This influences the likelihood of the document having the same topic in a lesser degree (more clusters mean more cases when the next document must be determined but all documents of the cluster are already taken).

Another problem is we cannot use Rand index to compare the clustering solution with the MLT solution described below (Chapter 6).

We introduce a similarity score to address both problems mentioned above. Similarity score is the average probability that the next document in the stream has the same topic as the current. We calculate the Similarity Score empirically by running the process several times and averaging the results. In case of clustering we select randomly the cluster and then select randomly samples inside the cluster. In case of MLT we select randomly the first document and then use MLT search to find the most similar document out of all documents not handled yet. If MLT search returns nothing the next document is selected randomly.

While doing this we calculate for each topic the frequency

$$F_t = \frac{S_t^{poz}}{S_t} \quad (1)$$

Here F_t is the Similarity score; S_t^{poz} is a number of times when the document of topic t is followed by the document of the same topic, S_t is a total number of documents of topic t.

To evaluate document clustering we calculate F_t for the largest topics (Figure 4). Similarity score for the two largest topics as well as the average total of the clustered items of the two topics are shown here for combinations of embedding, feature reduction and clustering methods. Configuration combinations are ordered here descending by the values of similarity score for the largest topic. The leftmost y-axis shows values of the similarity score while the rightmost – the total number of clustered documents by the configuration combination.

We see here that using truncatedSVD in concert with hdbscan reduces the amount of the clustered items (i.e. hdbscan creates a large number of outliers). It should be noted as well that tfidf is a good candidate as an embedding method.

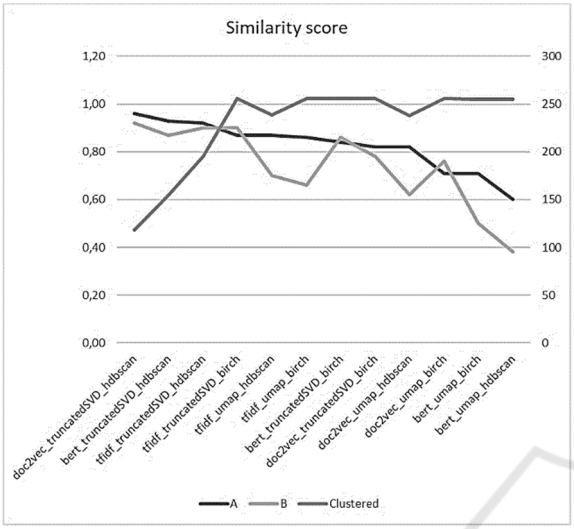

Figure 4: Similarity score.

6 MORE LIKE THIS QUERY

In Elasticsearch data store documents are stored as term-frequency vectors and the document frequency is precalculated at the index time for each term. This allows the fast extraction of term-to-term frequencies. Elasticsearch indexing mechanism allows as well to determine important terms using index data and standard tfidf procedure.

MLT searches for documents similar to a set of seed documents. In order to do so MLT selects a set of important terms of the seed documents and executes a query on those terms.

MLT has a number of configurable parameters that may influence the search result. To compare MLT performance for different combinations of parameters we executed MLT on the same sample set as used for the analysis of document clustering. For all largest topics we recorded the number of times MLT succeeded / failed to find the document of the same topic. This allowed to calculate frequency F_t of the next document to be of the same topic as seed documents.

One of the important MLT hyperparameters is the set of seed documents. If we select documents of a particular topic as seed documents than MLT may select a set of important keywords of the respective topic. This allows to retrieve documents with a similar set of important keywords. It is a good chance the retrieved documents belong to the topic of the seed documents.

Important issue in respect to the seed documents is – should we take all known documents of the topic as the seed? This may impact the query performance in case if the topic has a lot of documents (e.g. executing MLT on a set of 20 seed documents may take 4 times longer compared to a set of 2 documents even on a relatively small sample set). We should consider as well that the wording of topic documents may change gradually over the time. In order to address both those issues we introduce in our model hyperparameter $SEED_{max}$ and use at most $SEED_{max}$ newest documents as a seed for MLT. Results of experiments with various values of $SEED_{max}$ and some important standard MLT hyperparameters are outlined in Table 1.

Table 1: Important MLT hyperparameters.

Parameter	Description	Best values
$SEED_{max}$	Maximum seed documents.	2 – 3
max_query_terms	The maximum number of query terms used for MLT.	20 – 25
min_term_freq	The minimum term frequency below which the terms will be ignored.	1 – 3

Figure 5 shows comparison of similarity score for clustering and MLT. A combination of tfidf embedding, truncatedSVD feature reduction and birch clustering methods were used for clustering process as one of top performing combinations. $SEED_{max}$ 10, max_query_terms 25 and min_term_freq 2 were selected for MLT search. The Figure shows similarity scores for the 4 largest topics (presented on x-axis with the topic name and count of topic's documents). The similarity score for MLT and clustering is shown as well as share of the topic in the total document set. Standard deviation for averages is less than 7% for the MLT search and 9% for the clustering.

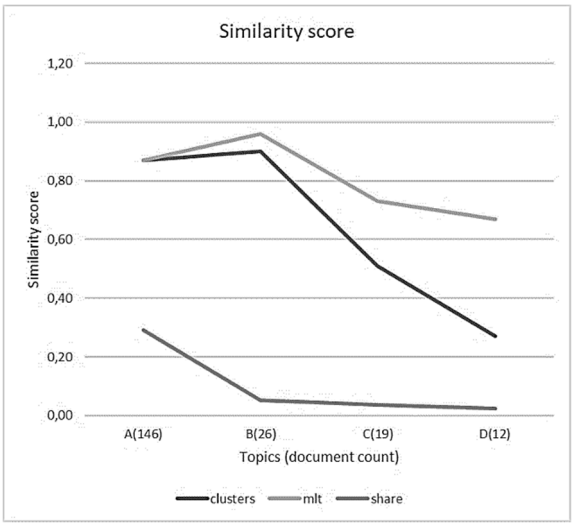

Figure 5: Comparing similarity score for clustering and MLT search.

No surprise that similarity scores for smaller topics are lower than those for larger topics. The diagram is here to compare results for MLT search and clustering for each particular topic.

As we can see both MLT and clustering improve the probability that the next document comes from the same topic (as compared to probability for random selecting that equals to the topic share). Elasticsearch MLT provides better results though as the explored clustering methods. This means that determining on the fly a closest match for the current text gives a better chance to get the document of the same topic in contrary to clustering documents in advance and then selecting the next document from the same cluster. The possible reason might be that unsupervised learning (and clustering in particular) has no knowledge about what terms are important to distinguish topics of interest. This is exactly the knowledge that supervised learning methods gain from the annotated trainsets.

Another observation here is that the similarity score depends on the topic. The possible reason is that some topics are better separated. E.g. topic B is better separated from other topics than topic D. Further research is necessary here to understand if this information can be used to improve topics (i.e. merge, split or redefine some topics,).

7 CONCLUSIONS

A model for automated handling of incoming enterprise documents is introduced in this article. The model comprises two processes – setup and main process. The goal of the setup process is a creation of annotated trainsets for classification bots of the main process. Applying clustering methods and MLT to support the annotation process is explored. We assume that grouping stream of documents presented to experts in clusters of similar documents should improve both the process of topic selection and that of document annotation. We assume as well that the configuration of the solution (both clustering and MLT) is better if higher the likelihood the next document in the stream is of the same topic as the current one. The similarity score metric introduced here (section 5.2) thus allows to compare solutions.

As results of the experiments show both clustering and MLT may be used to improve the annotation process of our model. It appears as well that MLT performs better than clustering here. This means that MLT is a viable option for the support of trainset annotation for text classification of enterprise documents.

Further research is feasible in several directions. It should be explored if it is possible to use data of the topic's documents (e.g. clustering data) to suggest topic improvements (e.g. merging, splitting or reorganizing topics). Another area for development would be to use paragraphs or sentences as the main items for topic prediction (and for annotation).

ACKNOWLEDGEMENTS

The research accounted for in this paper is co-funded by the European Regional Development Fund (ERDF, project No. 1.2.1.1/18/A/003, research No. 1.17).

REFERENCES

Assent, I., & Seidl, T. (2009). *Evaluating Clustering in Subspace Projections of High Dimensional Data*. http://dme.rwth-aachen.de/OpenSubspace/evaluation

Avinash M, & Sivasankar E. (2019). *A Study of Feature Extraction techniques for Sentiment Analysis*. 1–12.

Balakrishnama, S., & Ganapathiraju, A. (1998). *Linear Discriminant Analysis - a Brief Tutorial*. *January*, 8.

Bayyapu, K. R., & Dolog, P. (2010). Tag and Neighbour Based Recommender System for Medical Events. *Proceedings of the First International Workshop on Web Science and Information Exchange in the Medical Web, MedEx 2010*, 14–24.

Cygan, N. (2021). *Sentence-BERT for Interpretable Topic Modeling in Web Browsing Data Stanford CS224N Custom Project.*

Dinh, D., & Tamine, L. (2012). Towards a context sensitive approach to searching information based on domain specific knowledge sources. *Journal of Web Semantics*, *12–13*, 41–52. https://doi.org/10.1016/J.WEBSEM.2011.11.009

Fu, M., Qu, H., Huang, L., & Lu, L. (2018). Bag of meta-words: A novel method to represent document for the sentiment classification. *Expert Systems with Applications*, *113*, 33–43. https://doi.org/10.1016/J.ESWA.2018.06.052

Fu, Y., Feng, Y., & Cunningham, J. P. (2020). Paraphrase Generation with Latent Bag of Words. *ArHiv*.

Hansen, P. C. (1987). The truncatedSVD as a method for regularization. *BIT Numerical Mathematics 1987 27:4*, *27*(4), 534–553. https://doi.org/10.1007/BF01937276

Harris, Z. S. (1954). Distributional Structure. *Distributional Structure, WORD*, *10*(3), 146–162. https://doi.org/10.1080/00437956.1954.11659520

Jacobi, C., Van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, *4*(1), 89–106. https://doi.org/10.1080/21670811.2015.1093271

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text. *Int l Conf on Machine Learning (ICML). Ten-7301*, *9*.

Kadriu, A., Abazi, L., & Abazi, H. (2019). Albanian Text Classification: Bag of Words Model and Word Analogies. *Business Systems Research Journal*, *10*(1), 74–87. https://doi.org/10.2478/bsrj-2019-0006

Karpov, I., & Goroslavskiy, A. (2012). *Application of BIRCH to text clustering*. https://www.researchgate.net/publication/286672732_Application_of_BIRCH_to_text_clustering

Klinger, J. (2019). *Big, fast human-in-the-loop NLP with Elasticsearch | by Joel Klinger | Towards Data Science*. https://towardsdatascience.com/big-fast-nlp-with-elasticsearch-72ffd7ef8f2e

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). Text Classification Algorithms: A Survey. *Information (Switzerland)*, *10*(4). https://doi.org/10.3390/info10040150

Le, Q. V., & Mikolov, T. (2014). *Distributed Representations of Sentences and Documents*.

Liu, Q., Kusner, M. J., & Blunsom, P. (2020). *A Survey on Contextual Embeddings*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). *Learning Word Vectors for Sentiment Analysis*.

Malzer, C., & Baum, M. (2019). A Hybrid Approach To Hierarchical Density-based Cluster Selection. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, *2020-Septe*, 223–228. https://doi.org/10.1109/MFI49285.2020.9235263

McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. https://doi.org/10.48550/arxiv.1802.03426

Mikolov, T., Chen, K., Corrado, G. S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ICLR*, 1–12.

Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis: Foundations and Trends in Information Retrieval*. *2*(1–2), 1–135. https://doi.org/10.1561/1500000011

Parsons, L., Haque, E., & Liu, H. (2004). Subspace Clustering for High Dimensional Data: A Review *. *Sigkdd Exporations*, *6*(1), 90–105.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Shawon, A., Zuhori, S. T., Mahmud, F., & Rahman, J. (2018). Website Classification Using Word Based Multiple N-Gram Models And Random Search Oriented Feature Parameters. *2018 21st International Conference of Computer and Information Technology (ICCIT)*, *21-23 December*, 1–6. https://doi.org/10.1109/ICCITECHN.2018.8631907

Steinbach, M., Kumar, V., & Ertöz, L. (2003). *The Challenges of Clustering High Dimensional Data Big Data in Climate View project Understanding Climate Change: A Data Driven Approach View project The Challenges of Clustering High Dimensional Data *. https://doi.org/10.1007/978-3-662-08968-2_16

Taloba, A. I., Eisa, D. A., & Ismail, S. S. I. (2018). *A Comparative Study on using Principle Component Analysis with Different Text Classifiers*.

Tong, Z., & Zhang, H. (2016). *A Text Mining Research Based on LDA Topic Modelling*. 201–210. https://doi.org/10.5121/csit.2016.60616

Ular, M., & Robnik-Šikonja, M. (2020). High quality ELMo embeddings for seven less-resourced languages. *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, 4731–4738. http://hdl.handle.net/11356/1064

Vola, S. (2017). *How to use ElasticSearch for Natural Language Processing and Text Mining — Part 2 - Dataconomy*. https://dataconomy.com/2017/05/use-elasticsearch-nlp-text-mining-part-2/

Warrens, M. J., & van der Hoef, H. (2020). Understanding the rand index. *Studies in Classification, Data Analysis, and Knowledge Organization*, 301–313. https://doi.org/10.1007/978-981-15-3311-2_24

Yellai, M. (2016). *GitHub - pandastrike/bayzee: Text classification using Naive Bayes and Elasticsearch*. https://github.com/pandastrike/bayzee

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, *25*(2), 103–114. https://doi.org/10.1145/235968.233324