A comparison on the classification of short-text documents using Latent Dirichlet Allocation and Formal Concept Analysis

Noel Rogers, Luca Longo*

School of Computing, Dublin Institute of Technology, Ireland
luca.longo@dit.ie

Abstract. With the increasing amounts of textual data being collected online, automated text classification techniques are becoming increasingly important. However, a lot of this data is in the form of short-text with just a handful of terms per document (e.g. Text messages, tweets or Facebook posts). This data is generally too sparse and noisy to obtain satisfactory classification. Two techniques which aim to alleviate this problem are Latent Dirichlet Allocation (LDA) and Formal Concept Analysis (FCA). Both techniques have been shown to improve the performance of short-text classification by reducing the sparsity of the input data. The relative performance of classifiers that have been enhanced using each technique has not been directly compared so, to address this issue, this work presents an experiment to compare them, using supervised models. It has shown that FCA leads to a much higher degree of correlation among terms than LDA and initially gives lower classification accuracy. However, once a subset of features is selected for training, the FCA models can outperform those trained on LDA expanded data.

1 Introduction

In recent years the amount of short text data available online has exploded. A big part of this is down to the rise of social media with a lot of this data taking the form of tweets, Facebook posts or comments on media sites like YouTube for example. However, the sparse, noisy nature of short text makes automatic classification a difficult task. Typically a classifier could take tf-idf as inputs in the form of a Term-Document-Matrix (TDM) where entry t_{ij} relates the frequency with which term j appears in document i with the overall occurrances of the term across the document corpus [19] but for short-text the amount of information contained in a TDM is too sparse to facilitate accurate prediction. As a result we need to reduce this level of sparsity by adding weights in the TDM for words which do not already appear in the document. This could be done by incorporating external knowledge bases [17] or by using metadata to add extra features to compensate for the sparsity within the actual text [21]. Both rely on data external to the textual content so as an alternative the co-occurrence of words within the document corpus can be used to perform the necessary expansion. Two such techniques which adopt this approach are Latent Dirichlet

Allocation (LDA) and Formal Concept Analysis (FCA). An investigation into the application of these two techniques to text classification will be the primary focus of this work.

The rest of this document is organised as follows. Firstly, a brief review of related literature is provided, with particular emphasis on the applications of LDA and FCA to the problem of short-text classification. Section 3 then outlines the design of an experiment with the aim of comparing the improvements in classification accuracy due to each technique. An analysis of the results of this experiment are provided before we finish the paper with conclusions drawn from these results and provide suggestions for future work.

2 Related Work

2.1 Latent Dirichlet Allocation

Latent semantic analysis was developed to find the latent topics in a set of documents by looking at eigenvectors and used these as a means of dimensionality reduction [20]. This was extended to instead use conditional probabilities as a means of modelling the underlying topics, first introduced in [12]. The key idea is that a document can be considered as a mixed distribution over a number of topics. So, supposing there are k possible topics, then the probability that a given word w will instantiate some term t, is given by

$$p\left(w=t\right)=\sum_{k}p\left(w=t\mid z=k\right)p\left(z=k\right) \tag{1}$$

By convention we denote $\phi^{k}=p\left(w\mid z=k\right)$ as the word distribution for a topic k, and $\theta^{d}=p\left(z\right)$ as the distribution over topics for a given document d. Combining the distributions for all values of k and d respectively yields two matrices denoted ϕ and θ. Generalising θ to new documents not in the original corpus is non-trivial, so an additional assumption was taken in the seminal work of Blei, Andrew and Jordan which introduced a Dirichlet prior, leading to LDA [3]. A Dirichlet distribution is simply a family of distributions parameterised by a vector, α, of real values. In the case of LDA, the family of distributions correspond to ϕ^{k} and the values of α can be thought of as a prior count on the number of times a topic k is observed in a document. The same Dirichlet assumption can be extended to the distributions of words within topics, parameterised by a vector β. LDA is a generative model - we can generate a document word by word by first randomly sampling from the topic distribution and then selecting a word, conditioned on the selected topic [23], [4]. To generate an LDA model Markov-chain Monte Carlo moethods such as Gibbs sampling can be employed, for a detailed example see [23]. For the model parameters, the number of topics that should be generated may be known in advance but typically there needs to be a way to find an optimum value. There is no hard and fast rule for this, though there are heuristics based on information theory such as measuring the perplexity

on a hold-out test sample and then finding the topic number that minimises this. Perplexity gives a measure of how well the model predicts the distribution on the test documents and is computed as per equation 2 where M is the number of documents in the test set, w_d represents document d and N_d is the number of words in document d [25].

$$perplexity = \exp\left(-\frac{\sum_{d=1}^{M} \log p\left(w_d\right)}{\sum_{d=1}^{M} N_d}\right) \tag{2}$$

There have been a large number of examples applying LDA to text classification problems, with Twitter proving a popular data source for focusing on short-text problems [13]. For other applications LDA is simply one step in a more complex workflow to aid in achieving high classification accuracies [16, 6].

2.2 Formal Concept Analysis

We provide here a very brief overview of the subject of FCA. For a more detailed introduction to the topic see [10]. FCA was born out of a mathematical attempt to add formal definitions and structure to the notion of a concept. Intuitively a concept is a unit of thought consisting of a set of objects belonging to it (Called the *extent*) and the properties or attributes that they share (The *intent*). To formally define these ideas, start with a set of objects, X, and a set of attributes, Y, pertaining to elements of X. A binary relation, I, encodes for which elements in X have particular attributes of Y. The notation $\langle x, y\rangle \in I$ means that the object x has the attribute y. The collection $\langle X, Y, I\rangle$ is called a **formal context** [2].

A **formal concept** then, is a pair $\langle A, B\rangle$ where $A \subseteq X$, $B \subseteq Y$ with $A = \{x \in X \mid \forall y \in B, \langle x, y\rangle \in I\}$ and $B = \{y \in Y \mid \forall x \in A, \langle x, y\rangle \in I\}$

The sets A and B are the extent and intent of the concept respectively. The collection of all such concepts for a given context $\langle X, Y, I\rangle$ is denoted by $\mathcal{B}\left(X, Y, I\right)$. By ordering concepts using sub / super-set relations a partial ordering can be added to the set of concepts. The key theorem, taken from the seminal paper of Wille which initially produced this framework, is that $\mathcal{B}\left(X, Y, I\right)$ forms a lattice when equipped with this partial ordering [24]. When applied to short-text classification, the typical approach is to treat documents as the objects and the words appearing within them as the attributes. In this way a corpus of documents can be mapped to a concept lattice to determine the relationships between words [18]. The most relevant work for this paper is that of Boutari, Carpetino and Nicolussi [5]. Here, FCA is used as a text expansion technique to improve both supervised and unsupervised classification of short texts. Their main focus is on identifying proximity measures between concepts in the lattice that can be used to expand a TDM with weights from closely related concepts. In order to formalise this the authors developed five different metrics to generate these weights with the resulting matrices used as the input to K-Nearest Neighbour and K-Means classifiers for comparison.

3 Experiment Design

The key focus of this study is on comparing **LDA** and **FCA** as sparsity reduction techniques. In order to determine their comparative performance, classifiers will need to be trained on inputs derived from each technique and their accuracies compared - for this both neural networks and SVM have been chosen. A baseline model will be trained on the unprocessed input TDM. The key steps are shown in figure 1.

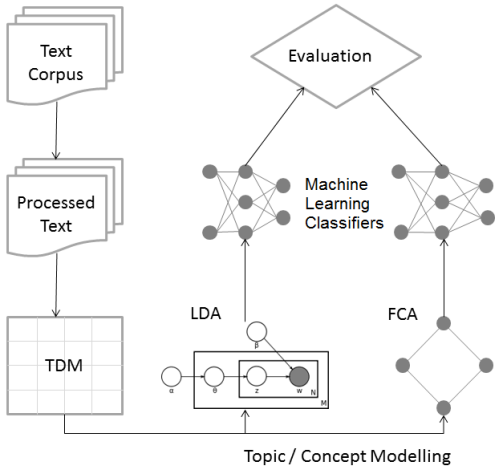


Fig. 1: Experiment design

3.1 Data and Preprocessing

To reduce the possibility of the specific patterns in the distribution of the dataset from impacting the results of the study, the experiment will be replicated using two distinct datasets. The first is the Google Snippets[1] corpus, first employed in [17]. It consists of snippets of search terms, typically between ten and forty words long, which comprise the documents. Each document is also assigned one of eight class labels. The dataset is already split into training and test subsets. The second dataset chosen is the Reuters-21578[2] collection. This is one of the most widely utilised datasets within the text classification domain, employed for example in [5] and [3]. This corpus consists of 21,578 different news articles along with additional metadata such as the author, date and title. For this study the articles themselves are too long so just the titles will be extracted with each

[1] jwebpro.sourceforge.net/data-web-snippets.tar.gz
[2] archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection

Table 1: Dataset Characteristics

(a) Google Snippets Dataset

Subset	Size	Min Len	Max Len	Avg. Len
Training	10,060	1	38	17.87
Test	2,280	2	38	17.96

(b) Reuters Dataset

Size	Min Len	Max Len	Avg. Len
7,733	2	36	6.48
3,561	2	43	6.47

considered a distinct document as per the approach taken in [5]. A number of recommended subsets and splits are included with the dataset, for the purposes of this experiment a subset will be taken consisting of 78 classes, a training set of 7,733 documents and a test set of 3,561. A summary of this is provided in table 1b. As pre-processing steps we first convert to lowercase and remove any punctuation or non alpha-numeric characters. All stop words then (For example "the" or "and") will be extracted. These do not contribute anything to the topics or concepts contained in a document so they will represent noise in the data. Note that the core focus of the experiment is on expansion of short text by enriching with topics or concepts derived from the entire corpus. Words which only appear in a single document are of little use in this regard, since they cannot form relationships with words from other documents. Therefore any words that only appear in a single document will be removed. Once this processing has been completed, a sparse TDM T is generated for each dataset. Each element t_{ij} represents the inverse-frequency with which word j appears in document i. LDA and FCA will be employed to address the sparsity of T.

3.2 LDA

The first point of note is that the LDA step can be tuned by a number of hyperparamaters - the Dirichlet priors α and β and the number of topics. Ideally a range of values would be tried for each so that the optimal value could be found but this will not be feasible for this study so a good approximation for each needs to be taken upfront. The approach taken will follow the same as that employed in [11]. Their recommendation is to take $\alpha = 50/N_T$ and $\beta = 200/W$ where N_T is the number of topics and W is the number of words. To derive the number of topics to use, perplexity values will be calculated from the test data as per equation 2. The range of values taken will be from 1 to 246, incrementing by 5 each time. Based on the outcome of this test a single model will be chosen to proceed with. The key outputs from this model are two probability matrices - one giving the distribution of words within each topic and the other giving the distribution of topics over documents. These correspond to ϕ and θ respectively, as defined in section 2.1. It follows then from equation 1 that the probabilities for each word appearing in each document are given by $\theta \times \phi$. This new matrix has the same dimensions as T and replaces T as the input for the training step.

3.3 FCA

The starting point is to note that T can be considered a formal context - if $t_{ij} \neq 0$ then word j appears in document i. As such we can form a concept lattice from the documents and words and related concepts from this will be used to add non-zero terms to T. Once the concept lattice is formed a proximity measure can be derived, encoding how closely related two concepts are. For this we choose

$$Proximity = 1 - \frac{SD}{\max SD} \tag{3}$$

where SD is the shortest distance between two points in the graph [5]. Given any pair of words, equation 3 allows the similarity between them to be computed yeilding a symmetric matrix S where each term s_{ij} is the proximity between words i and j. Now let d be a vector representing one of the documents i.e. d corresponds to a row of T. The aim is to obtain a new representation, d', that takes advantage of the word proximities to reduce the sparsity of the original representation. The value for the i^{th} word should take into account both the proximity between word i and each other word but also the frequency with which those other words appear in d yielding the following equation

$$d'_i = \sum_{k=1}^{W} d_k S_{ki} \tag{4}$$

Extending this over the whole document gives $d' = d \times S$. It follows then that the expanded term document matrix, T', is simply $T \times S$.

For the execution of this we adopt a tool[3] implementing the InClose algorithm [1] to obtain the concept set. To construct the lattice from these concepts, a simple algorithm was employed to add edges where a given concept was a lower neighbour of another. The matrix S was generated using a breadth-first search over the lattice. Note that the source and sink nodes (Corresponding to the empty and universal sets in the concept) need to be first removed so that concepts cannot be linked via these nodes. It could happen that two unrelated terms end up with an unnaturally high proximity value on account of both being directly connected to either the source or sink.

3.4 Modelling

We briefly highlight the choices made for neural network parameters. As we have a clasification problem the activation function selected is softmax. For the hidden layer, rectified linear units or ReLU will be used [14]. In order to help the model generalise well and avoid overfitting, dropout layers will be added in between each layer of the network [22]. The final consideration is to the architecture of the model - the number and width of hidden layers and the connections between them. Since the purpose of this study is not to investigate

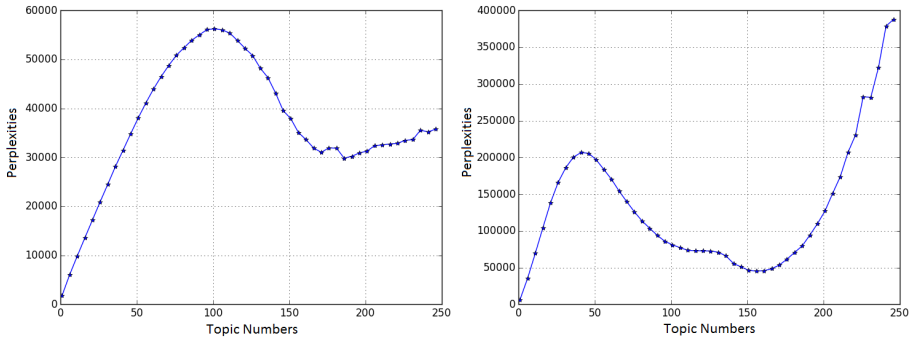
[3] sourceforge.net/projects/inclose/

neural network architectures, the simplest setup will be chosen, namely a single hidden layer, of width W and with all units connected. One additional pre-processing step that will be required before training the neural network is to normalise the input data. The normalised features, or z scores, are computed by subracting the mean and dividing by the std. deviation of each feature. For the SVM, a simple linear kernel has been chosen. The only other consideration required with the algorithm then is on how to deal with the multiple class labels. In order to handle this the 'one-versus-rest' approach will be taken [15].

4 Results and analysis

4.1 LDA

The topic values taken for each LDA model were determined by computing the perplexity associated to each topic number and selecting the minimum. From figures 2a and 2b these topic numbers are 181 and 161.



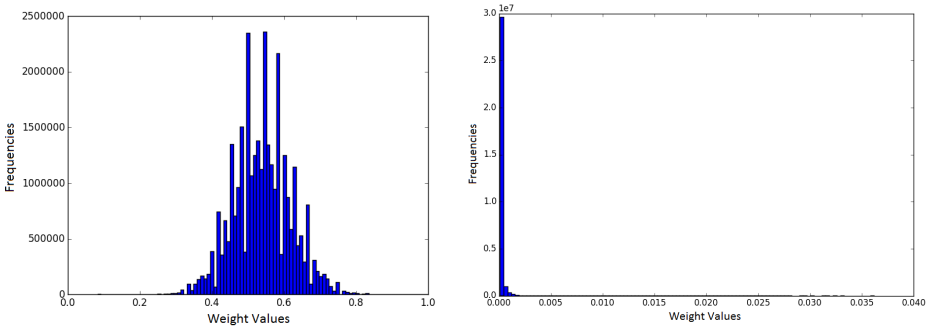
(a) Reuters perplexity values (b) Snippets perplexity values

Fig. 2: Perplexity values per topic number

4.2 Correlation Analysis

The initial experiments yielded very poor accuracy on the FCA enhanced Snippets dataset for neural networks and it was found that there was a high degree of correlation between the input features. As a result of this two further runs of the experiment were performed, the first removing correlated features above a threshold of 0.8. This conservative value gave little improvement and prompted a further run where the top 10% of features were selected based on the outcome of an ANOVA test.

To help understand the cause of the high correlations, the distributions of weights in the FCA and LDA enhanced TDMs for the Snippets dataset are shown (Figures 3a and 3b). For the LDA weights, the majority of terms are close to zero so it is still just a small subset of terms that contribute most to the classification. Contrast this with FCA; here the weights form a near normal distribution around a mean of 0.5. The impact is that even totally unrelated terms are still contributing to significant weight increases. Across both datasets, the greatest distance between any pair of concepts was 12 leading to a small range of values that the proximities could take. We will revist this issue in section 5 with suggestion for how future work can combat this problem.



(a) FCA TDM weights (b) LDA TDM weights

Fig. 3: Weight distributions

4.3 Overall model evaluation

Neural Network models The performance of each classifier was determined by comparing precision, recall and F-measure values (Denoted P, R and F_1). A full breakdown of the results of each run of the experiment are given in tables 2a to 2c. Graphs of the F_1 scores can also be seen in figures 4a and 4b for the Reuters and Snippets experiments respectively. As already highlighted, the initial FCA results are quite poor on the Snippets dataset. For Reuters however, FCA is already outperforming both the baseline (BL) and LDA. Removing correlated features does not lead to significant change in the results 2b but in the final run of the experiment, following the selection of just 10% of features using ANOVA, it can be seen that FCA has outperformed across the board.

SVM models The high correlations which impeded the FCA trained neural networks did not have the same negative effect on the SVM models. Across both datasets the highest accuracies are achieved on the first run, before features are removed. Comparing FCA and LDA for the SVM models, the highest overall F_1

Table 2: Experimental Results

		Reuters			Snippets		
		BL	FCA	LDA	BL	FCA	LDA
NN	P	0.715	**0.78**	0.59	**0.645**	0.017	0.6
	R	0.718	**0.73**	0.59	**0.61**	0.13	0.58
	F_1	0.706	**0.74**	0.57	**0.609**	0.03	0.58
SVM	P	0.68	**0.78**	0.52	0.61	**0.72**	0.58
	R	0.68	**0.78**	0.58	0.57	**0.69**	0.56
	F_1	0.67	**0.78**	0.54	0.57	**0.69**	0.56

(a) Results - No feature engineering

		Reuters			Snippets		
		BL	FCA	LDA	BL	FCA	LDA
NN	P	0.736	**0.78**	0.62	**0.656**	0.021	0.63
	R	0.736	**0.76**	0.59	**0.622**	0.14	0.59
	F_1	0.724	**0.76**	0.59	**0.621**	0.04	0.59
SVM	P	0.68	**0.78**	0.55	0.61	**0.71**	0.6
	R	0.68	**0.77**	0.59	0.57	**0.68**	0.58
	F_1	0.68	**0.77**	0.55	0.56	**0.68**	0.58

(b) Results - Correlation removal

		Reuters			Snippets		
		BL	FCA	LDA	BL	FCA	LDA
NN	P	0.698	**0.77**	0.61	0.631	**0.72**	0.62
	R	0.693	**0.78**	0.62	0.551	**0.66**	0.61
	F_1	0.681	**0.76**	0.61	0.555	**0.66**	0.61
SVM	P	0.66	**0.74**	0.56	0.57	**0.65**	0.64
	R	0.6	**0.75**	0.61	0.52	**0.63**	0.62
	F_1	0.57	**0.74**	0.56	0.52	**0.63**	0.62

(c) Results - ANOVA feature selection

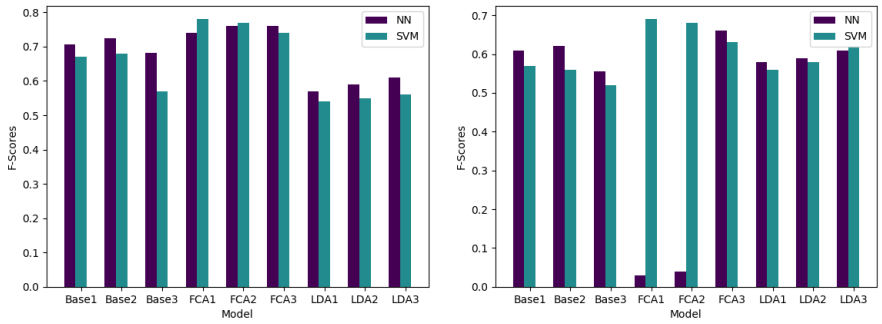
values are again achieved by FCA (0.69 versus 0.62 on the snippets data and 0.78 versus 0.56 on Reuters). Comparing the best F_1 scores for each dataset across all 3 runs shows the FCA achieving a 3-5% increase on the baseline and a 5-15% increase over LDA. The highest scoring combination across both datasets is FCA + SVM with no need for additional feature engineering steps.

4.4 Statistical significance

As a final point the statistical significance of the the obtained results has been evaluated. We used McNemars test statistic [8]. The statistic is given by

$$\chi^2 = \frac{(\mid n_{01} - n_{10} \mid -1)^2}{n_{01} + n_{10}} \tag{5}$$

where, for two models a and b, n_{01} corresponds to cases misclassified by a and not b and those missed by b and not a give n_{10}. From table 3, it can be seen that the results comparing the LDA and FCA models are statistically significant with a p-value < 0.01.



(a) F_1 scores for Reuters data

(b) F_1 scores for Snippets data

Fig. 4: Model evaluation using F_1 scores

	Snippets		Reuters	
	NN	SVM	NN	SVM
McNemar χ^2	13.62	91.72	10.32	9.11
P-value	< 0.01	< 0.01	< 0.01	< 0.01

Table 3: Statistical significance of experimental results

4.5 Summary of results

Three different iterations were run - without feature engineering, with the removal of correlated features and incorporating ANOVA for feature selection. Two classifiers were trained on the resulting feature sets - neural networks and SVM, with FCA showing a 5% increase in the Snippets dataset and a 15% improvement on the Reuters data. The LDA models remained consistent throughout but failed to even outperform the baseline models on either dataset. Analysis was performed to understand the initial poor scores and a high degree of feature correlation was discovered. As the focus was on term expansion techniques, no parameter engineering was performed on any of the neural network models and only a simple linear kernel was employed for the SVM. Varying the network architectures, dropout weights or learning rates or employing more sophisticated kernels could have improved the results for individual models but these steps were not performed.

To strengthen the results the experiment was repeated on two datasets. The correlation problem that FCA initially introduced may not have been picked up had only the Reuters dataset been applied. LDA is widely used in text analysis but we have shown that for this particular task FCA is more suitable. We have also shown though that FCA also adds a high degree of correlation between terms. One of the drawbacks in FCA is the computational resources needed to

build the concept lattice and term similarity matrix. The density of the lattice was highlighted as a reason for the high correlations so a trade-off in computing the full lattice or term similarities would help mitigate both the resources required and reduce the correlations.

5 Conclusions

This experiment has compared the relative benefits of LDA and FCA for to addressing the sparsity of short-text. We list now some potential future avenues of work arising from this experiment. This project only focused on "standard" FCA, however fuzzy FCA, as described in [7], could be examined. In fuzzy FCA, rather than attributes simply being absent / present for a given object, a weight between 0 and 1 is applied to each one - precisely the form that a tf-idf TDM takes. The outcome of the FCA model is the term-term similarity matrix and this is the key component in this step. One measure was utilised in this work, however there are alternative methods of deriving concept similarity from a lattice, not just the geometric distance of shortest paths. Further work could look at alternatives such as set based approaches (Measuring the size of the intent / extent intersections) or combinations of these with geometric distance [5]. One issue identified with FCA was the high degree of correlations that were observed. We looked at evaluating term-similarities between a concept and all other others but a more restrictive approach, looking just at a small neighbourhood around each concept, might fare better. Within this neighbourhood the proximities could be computed as before, with all concepts outside this neighbourhood having being set to 0 [9]. An alternative approach that could yield the same outcome is to instead use *iceberg lattices* [1]. This is simply a concept lattice which has been pruned by introducing a required minimal support for concept inclusion. The removal of edges from the lattice would lead to a wider spread in proximities between concepts.

References

1. Andrews, S., Hirsch, L.: A Tool for Creating and Visualising Formal Concept Trees. In: Proceedings of the Fifth Conceptual Structures Tools & Interoperability Workshop. pp. 1–9. Annecy, France (2016)
2. Belohlavek, R.: Introduction to formal concept analysis. Palacky University, Department of Computer Science, Olomouc p. 47 (2008)
3. Blei, D., Andrew, Y., Jordan, M.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1020 (2003)
4. Blei, D., Carin, L., Dunson, D.: Probabilistic topic models. IEEE Signal Processing Magazine 27(6), 55–65 (2010)
5. Boutari, A.M., Carpineto, C., Nicolussi, R.: Evaluating term concept association measures for short text expansion: Two case studies of classification and clustering. CEUR Workshop Proceedings 672, 163–174 (2010)
6. Chen, M., Jin, X., Shen, D.: Short text classification improved by learning multi-granularity topics. In: IJCAI. pp. 1776–1781 (2011)

7. De Maio, C., Fenza, G., Loia, V., Senatore, S.: Hierarchical web resources retrieval by exploiting fuzzy formal concept analysis. Information Processing and Management 48(3), 399–418 (2012)
8. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10(7), 1895–1923 (1998)
9. Eklund, P., Ducrou, J., Dau, F.: Concept similarity and related categories in information retrieval using formal concept analysis. International Journal of General Systems 41(8), 826–846 (2012)
10. Ganter, B.: Ch1 & Ch2: Contexts, Concepts, and Concept Lattices. Formal Concept Analysis: Methods and Applications in Computer Science (2003)
11. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America 101 Suppl, 5228–35 (2004)
12. Hofmann, T.: Probabilistic latent semantic indexing. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval pp. 50–57 (1999)
13. Hong, L., Davison, B.: Empirical study of topic modeling in twitter. Proceedings of the first workshop on social media analytics pp. 80–88 (2010)
14. Kelleher, J.D., MacNamee, B., D'Arcy, A.: Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. The MIT Press (2015)
15. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: a stepwise procedure for building and training a neural network. Neurocomputing: algorithms, architectures and applications 68(41-50), 71 (1990)
16. Lu, Z., Li, H.: A Deep Architecture for Matching Short Texts. Advances in Neural Information Processing Systems pp. 1–9 (2013)
17. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. Proceeding of the 17th international conference on World Wide Web - WWW '08 pp. 91–100 (2008)
18. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Systems with Applications 40(16), 6538–6560 (2013)
19. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) 34(1), 1–47 (2002)
20. Song, W., Park, S.C.: Genetic algorithm for text clustering based on latent semantic indexing. Computers and Mathematics with Applications 57(11-12), 1901–1907 (2009)
21. Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short Text Classification in Twitter to Improve Information Filtering. Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval SE - SIGIR '10 pp. 841–842 (2010)
22. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: prevent NN from overfitting. Journal of Machine Learning Research 15, 1929–1958 (2014)
23. Steyvers, M., Griffiths, T.: Latent semantic analysis: a road to meaning, chapter probabilistic topic models. Laurence Erlbaum (2007)
24. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ordered sets, pp. 445–470. Springer (1982)
25. Zhao, W., Chen, J.J., Perkins, R., Liu, Z., Ge, W., Ding, Y., Zou, W.: A heuristic approach to determine an appropriate number of topics in topic modeling. BMC Bioinformatics 16(Suppl 13), S8 (2015)