The Effect of Preprocessing on Short Document Clustering

Cynthia Koopman and Adalbert Wilhelm

Abstract Natural Language Processing has become a common tool to extract relevant information from unstructured data. Messages in social media, customer reviews, and military messages are all very short and therefore harder to handle than longer texts. Document clustering is essential in gaining insight from these unlabeled texts and is typically performed after some preprocessing steps. Preprocessing often removes words. This can become risky in short texts, where the main message is made of only a few words. The effect of preprocessing and feature extraction on these short documents is therefore analyzed in this paper. Six different levels of text normalization are combined with four different feature extraction methods. These setting are all applied on K-means clustering and tested on three different datasets. Anticipated results can not be concluded, however other findings are insightful in terms of the connection between text cleaning and feature extraction.

Cynthia Koopman
Jacobs University Bremen, Campus Ring 1 Bremen Germany,
✉ c.koopman@zeelandnet.nl

Adalbert F.X. Wilhelm
Jacobs University Bremen, Campus Ring 1 Bremen Germany
✉ a.wilhelm@jacobs-university.de

Archives of Data Science, Series A
(Online First)
KIT Scientific Publishing
Vol. 6, No. 1, 2020

DOI: 10.5445/KSP/1000098011/01
ISSN 2363-9881

1 Introduction

Data has rapidly become a very important source of information. Natural Language Processing (NLP) plays a big role in this advancement as it allows to gain information from unstructured data. This data, text documents, is in many applications very short, for example in tweets, customer reviews or even military messages. However, these short messages can contain important information about for example recent trends or customer specific information, which can be very beneficial for companies (Fan and Gordon, 2014). Working with short text documents is however different from working with longer texts. There is less context in general, and, therefore, it is harder to capture the main message. The volume of these messages is also large, too large for manual inspection.

In order to gain insights from these large volumes of unstructured data, which is almost never labeled, unsupervised learning techniques such as clustering become essential. Clustering could provide the requested insight by grouping similar documents together and with this providing an overview of the main findings of thousands of reviews in a concise manner (Fan and Gordon, 2014).

Methods like preprocessing and feature extraction in clustering become very important for short text documents because every word in these texts is of great importance. Texts like tweets or customer reviews are sparse, tweets have a maximum of 140 characters, but they also include many spelling mistakes, usage of slang and connected words (Crockett et al, 2017). This paper will therefore investigate the combination of text cleaning and feature extraction on short text document clustering.

2 Related Work

Several studies have been performed on clustering or text mining focused on short documents, these studies however often have limitations due to research problems being poorly defined (Ellen, 2011). There are many studies that propose new algorithms that work better but are often very specific, for example focusing only on Twitter data such as Poomagal et al (2015) or Adel et al (2014). Further, the data sizes used by studies are often not sufficient to gain insight from the sparse texts, for example Adel et al (2014) have a maximum data size of 5000 and Liu et al (2005) have an average data size of 1422 with more than 10 clusters. Studies may have good results, which are potentially related to the small data size which then leads to a lack in scalability. Also, preprocessing steps are often not evaluated extensively and this leads to documents that include redundant data and can develop inaccurate clustering. Liu et al (2005) have for example very high accuracy scores with their small data size. In the research of Rosa et al (2011) preprocessing steps were inspected in more detail, and resulted in observable differences of these levels. However, their research is solely focused on Twitter data and does not combine the experiment with feature extraction techniques. Evaluation metrics are also important to consider, and the previously mentioned studies often use only metrics from a specific field, for example Rosa et al (2011) only uses metrics based on statistics. Another similar study to the proposed study is by Tang et al (2005). They compare several different feature extraction methods on different data sets and have again very good scores, but very small datasets. The combination of cleaning text and feature extraction could however be fundamental since some feature extraction methods work more with context than others, and removing too much information in the cleaning step could give low or inaccurate results (Allahyari et al, 2017). An important study in this context is done by Uysal and Gunal (2014), they compared preprocessing steps with many aspects of supervised learning including feature extraction and found that it makes a significant difference in text classification. However, with popular advances in feature extraction techniques, this study lacks the comparison of these new techniques and focuses more on language dependencies for example.

This research will focus on the preprocessing steps that come before clustering. Several preprocessing steps will be tested on different feature extraction techniques which together are necessary to remove redundant data and decrease dimensionality while still being able to capture the information from the data. This will be tested on data that comes from different sources with relatively large data size. The remainder of this paper is organized as follows: Section 3 will discuss all methods used, sect. 4 will present the results, sect. 5 will discuss the results and finally the paper will be concluded in sect. 6.

3 Method

In this section all preprocessing and feature extraction techniques are explained followed by a brief description of the clustering algorithm and data used. Finally, the evaluation metrics are presented.

3.1 Preprocessing

As a first step, text is cleaned before extracting features. There are many possibilities on how to normalize text, and it is often said that this depends on data. In texts which features many URLs for example, removing these could definitely make a difference, whereas in others not. However, there are speculations that it depends on the data only to a certain extend (Allahyari et al, 2017) (Tang et al, 2005). This research focuses rather on the common effects and will not take such outliers as URLs into consideration. For this research 6 different levels of normalization were established with increasing complexity. The levels are the following:

0. No normalization applied.
1. All texts to lowercase and punctuation removed.
2. Level 1 and removal of most common words and very rare words.
3. Level 2 and stop word removal.
4. Level 3 and stemming.
5. Level 3 and lemmatization.

For stop word removal the nltk library in Python is used, this removes all stopwords of the English language. Stemming is performed using the Porter Stemmer algorithm from nltk (Porter, 1980). Stemming removes a suffix, for example, "connected", "connection" and "connecting" will all be reduced to "connect" (Porter, 1980). The porter stemmer is often used since it is simple and fast. This can however not reduce conjugations to its root and will create words that are not considered words in english, for example "flies" is reduced to "fli". Lemmatization can take these limitations into account, and is therefore also included as a level of normalization using the wordNet lemmatizer from the nltk library (Fellbaum, 1998). Lemmatization determines the lemma of a word, it can reduce "walking", "walked" and "walks" to the lemma "walk". Lemmatization is an algorithmic process that can take context into account using part of speech tagging which stemming does not. The wordNet lemmatizer makes use of the wordNet english lexical database which has all cognitive synonyms grouped together and linked to each other (Fellbaum, 1998). Further, the order of text cleaning is also very important. In this study, the order is according to the level descriptions. Although lemmatization already works without putting all words to lowercase, this was done to remove common, rare, and stopwords correctly. Stopwords are removed always as a last step since some words may have been in a different form before.

3.2 Feature Extraction Methods

In order to use the text documents for clustering, they have to be transformed into numerical data. Four important feature extraction techniques were chosen.

The first technique is TFIDF, which is composed of multiplying Term Frequency (TF) by Inverse Document Frequency (IDF). This technique is included due its high popularity and intuitiveness (Robertson, 2004). TF, denoted by $f(t, d)$ is defined as the number of times term t occurs in document d. Document frequency, $df(t, d)$, is defined as the number of documents d that contain the term t. IDF follows as $IDF = log(\frac{N}{df(t,d)})$, where N denotes the total number of documents. Scikit learn (Pedregosa et al, 2011) is used to calculate TFIDF which includes an addition of 1 to IDF, such that terms with zero i.e. terms that occur in all documents do not get ignored completely. Further, zero division is avoided by an addition of 1 in the nominator and denominator. This results in

$$TFIDF = TF \cdot IDF = f(t, d) \cdot log(\frac{N + 1}{df(t, d) + 1}) + 1. \quad (1)$$

TFIDF essentially moves more importance to rarer words and, therefore, the higher the score, the rarer the word (Robertson, 2004). In order to reduce model complexity, a term limit was set for TFIDF. The combination of the large datasets and the numerous tests performed lead to this decision. The limit of terms was set to 5000 which was a reasonable amount to accommodate the time needed to perform the multiple experiments. The effect of this limitation will not influence the results as 5000 remains large and only extremely rare terms are excluded. However, a further drastic reduction could affect results as important words could be removed. Due to the need of repetition to reduce the randomness in the clustering method (see section 3.3), one experiment resulted in a computation time of about one hour on a 64 bit CPU. This limit indicates that only the top 5000 terms (ranked by their frequency) are used. The next feature extraction technique is derived from the first and has N-grams included. Different to the first technique now an TFIDF is built on pairs of words, bigrams and trigrams. Including N-grams is beneficial because it can preserve more context (Damashek, 1995).

Further, more complex word embeddings have become popular and two are used in this study: Word2Vec (W2V) and GloVe. Word2Vec uses a simple neural network to turn the documents into vectors and was first introduced in 2013 by Tomas Mikolov (Mikolov et al, 2013). Word2Vec is able to capture context and semantics in documents better than TFIDF. It is implemented using gensim (Řehůřek and Sojka, 2010). This Word2Vec implementation makes use of Common Bag of Words (CBOW). CBOW tries to predict the next word using only context and does this using shallow neural networks. A representation for every document is obtained by taking the average of the word vectors per document. GloVe (Global Vectors) is a pre-trained word embedding, a more powerful alternative to Word2Vec. GloVe is an algorithm introduced in 2014 that extracts features from text by using global matrix factorization and local context window methods (Pennington et al, 2014). GloVe offers several different vectors that can be used for word representation, in this research the GloVe embedding with 6 Billion tokens and 400k vocabulary was used (Pennington et al, 2014).

3.3 Clustering Algorithm

For every previously described setting, K-means clustering is performed. K-means clustering is used due to its simplicity such that the focus of this study remains on the preprocessing (Crockett et al, 2017). K-means also turned out to be faster in comparison to other models and is widely used for clustering (Allahyari et al, 2017). For implementation, Scikit learn (Pedregosa et al, 2011) is used and all settings are kept to default. An alternative would be to use Latent Dirichlet Allocation (LDA) (Blei et al, 2003) which is often used for topic clustering. This technique differs from k-means as it assigns a document to multiple clusters usually with probabilities. Nevertheless, k-means was chosen for this study because of the previously mentioned reasons. For future work, LDA would be an interesting alternative to k-means clustering. Due to K-means having a random component, all experiments were performed 10 times. Taking the mean of these 10 runs resulted in stable results, having very low standard deviations. The results discussed in the next sections are the mean of those 10 runs, the standard deviations are not investigated as these remained low.

3.4 The Data

All previous explained settings are applied on three datasets obtained from Amazon, Yelp and Wikipedia. All datasets have two classes such that one can compare results and capture the effects of preprocessing in an isolated manner.

The first dataset is composed from a larger dataset that contains Amazon reviews. Originally this dataset is obtained from the Stanford Network Analysis Project (SNAP) which contained 18 years of reviews (Leskovec and Krevl, 2014). This dataset was reduced by Zhang et al (2015) such that it contains the first two classes resulting in 10.000 documents in total. The labels correspond to either a positive or a negative review. The average number of words per document is 80.

The next dataset contains Yelp reviews and is again from a larger dataset. The dataset was earlier part of the Yelp Dataset Challenge in 2015 and contained almost two million reviews (Zhang et al, 2015). This was again reduced to two classes and resulted in 20.000 documents. It was reduced such that one class contained all reviews with one-star rating and the other class, those with a five-star rating. The average number of words per document is 127.

The final dataset was obtained through DBPedia, which is a crowd-sourced ontology dataset built from Wikipedia (Lehmann et al, 2015). Here two classes were chosen resulting in documents containing descriptions of a film or book. The resulting dataset had 10.000 documents. The average number of words per document is 47.

3.5 Evaluation Metrics

Using evaluation metrics for unsupervised learning is not straightforward and can be misleading. Therefore multiple metrics were used. The Adjusted Rand Index (ARI) and Adjusted Mutual Information (AMI) are used to measure how well the clusters are formed in a supervised manner. In terms of true/false positive and negative the Rand index (RI) and the ARI are defined as

$$RI = \frac{TP + TN}{TP + FN + TN + FP}, \qquad ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}. \tag{2}$$

The ARI has measures between -1 and 1, 1 being a perfect match, and it measures the similarity between clusterings with normalizing chance (Hubert and Arabie, 1985). The AMI measures the agreement between clusterings and is also normalized for chance (Vinh et al, 2010). It uses Mutual Information (MI) and they are defined as

$$MI = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{U_i \bigcap V_j}{N} log(\frac{N|U_i \bigcap V_j|}{|U_i||V_i|}), \tag{3}$$

$$AMI = \frac{MI - E[MI]}{mean(H(U), H(V)) - E[MI]}. \tag{4}$$

Where U and V are label assignments of N objects. AMI has the same measure bounds as ARI. ARI and AMI are similar metrics but have been found to work differently on different sizes of clusters and are therefore both included (Romano et al, 2016).

A metric for unsupervised learning is also used. The Adjusted Silhouette Width (ASW) (Rousseeuw, 1987) is used and is defined as

$$ASW = \frac{b - a}{max(a, b)}, \tag{5}$$

where a is the mean distance between a point and all other points in the cluster, and b is the mean distance between a point and all other points in the nearest cluster (Rousseeuw, 1987). ASW has measure bounds between -1 and 1, 1 being a highly dense clustering.

Besides these metrics, also the top 10 terms per cluster are investigated based on their interpretation. It should be noted that such an analysis is less objective than formal metrics, but can nevertheless provide more insight. The top terms per cluster were obtained by looking at terms closest to the centroid.

4 Results

All normalization levels, feature extraction methods and datasets sum up to 72 different models in total. Having three different evaluation metrics results in 216 metrics that need to be compared. The different settings for the experiment are summarized in Table 1. Due to this large number this section will only focus on certain results. The main objective is to find evidence that can generalize the effect of preprocessing. Therefore (dis)similarities are investigated.

Table 1: Experiment details.

Datasets	Normalization Levels	Feature Extraction Methods	Clustering Algorithm	Evaluation Metrics
	Level 0			
Amazon	Level 1	TFIDF		AMI
Yelp	Level 2	TFIDF N-gram	K-Means	ARI
DBPedia	Level 3	Word2Vec		ASW
	Level 4	GloVe		
	Level 5			

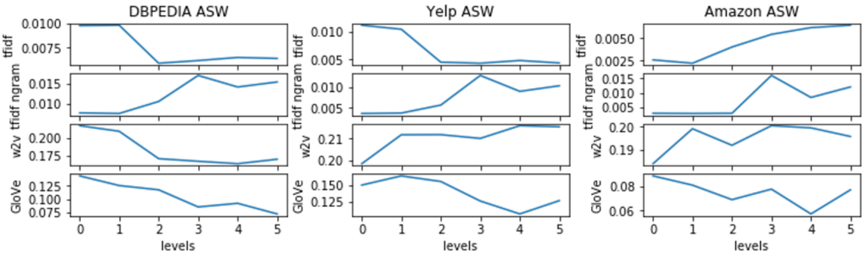

Figure 1: AWS for Yelp, Amazon and DBPedia for every feature extraction technique vs. normalization level.

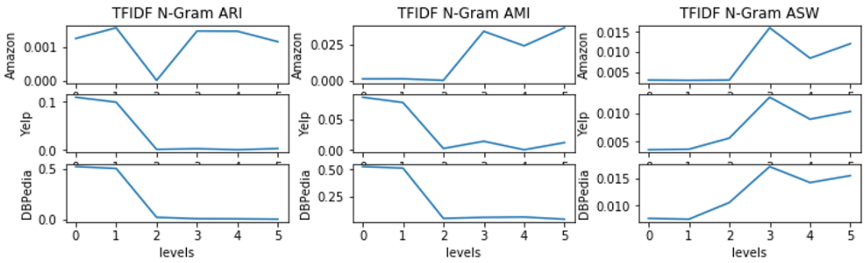

Figure 2: TFIDF N-gram ARI, AMI and ASW for Yelp, Amazon and DBPedia for every normalization level.

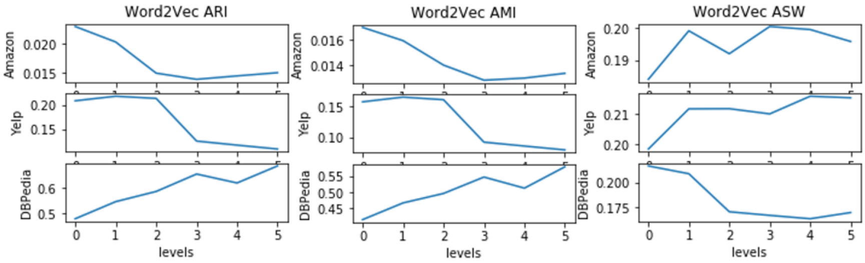

Figure 3: Word2Vec ARI, AMI and ASW for Yelp, Amazon and DBPedia for every normalization level.

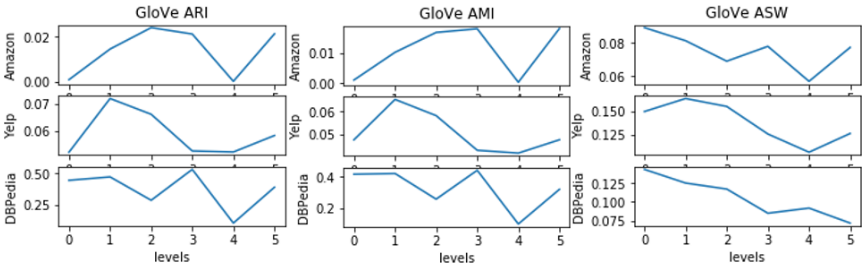

Figure 4: GloVe ARI, AMI and ASW for Yelp, Amazon and DBPedia for every normalization level.

Figure 1 shows the ASW for feature extraction method for each dataset and some similarities can be observed. When looking horizontally per feature extraction method, the path is almost equal in every dataset. The only exceptions are TFIDF in Amazon and Word2Vec in DBPedia.

Figure 2 provides an overview of all the TFIDF N-gram results for all datasets and evaluation metrics. The graph for the ASW shows the previous result more clearly. For ARI and AMI the same effect can not be seen however, the results in the Amazon dataset are different from the other datasets. Nevertheless, the result for AMI and ASW in the Amazon dataset are very similar. TFIDF N-gram is the only method where ARI and AMI are not identical.

Word2Vec shows also interesting similarities. Figure 3 shows that in Yelp and Amazon data the results for ARI and AMI are very similar. They have a big decrease until level 4 and only slightly change from then on.

The next figure focusses on GloVe. Figure 4 shows almost identical results per dataset in all metrics. This has not been observed in any other method. Another result that is best visible in this figure, is the decrease in level 4 (Stemming) and increase in level 5 (Lemmatization).

From the results it can be seen that in some experiments increasing normalization has adverse effects. This is however not observed consistently. Further, the best combination of feature extraction method and normalization level is analyzed. Table 2 gives an overview for all evaluation metrics and datasets. It shows that for maximizing ASW, Word2Vec is always best and for AMI a TFIDF based method is best. A homogeneous normalization level is not visible from these results, nevertheless levels 3 and 4 are the most common.

Table 2: Best combination of feature extraction method and normalization level, for all datasets and evaluation metrics.

	ASW	ARI	AMI
Amazon	W2V - level 3	GloVe - level 1	TFIDF Ngram - level 5
Yelp	W2V - level 4	W2V - level 2	TFIDF - level 3
DBPedia	W2V - level 0	TFIDF - level 4	TFIDF - level 4

Due to the difficulty in using appropriate evaluation metrics in unsupervised learning, an additional analysis was performed in terms of interpretation of the clusters. In total 12 pairs of 10 terms were compared. Comparing 6 different levels becomes difficult and therefore one can only determine an overall trend. For the Amazon dataset set an increasing trend is observed (except in Word2Vec) which is in line with earlier observations for the ASW. For the other datasets this is however not the case. Further, the most intuitive method for the Amazon and Yelp dataset was TFIDF N-gram. This was mostly due to the additional context provided by the N-grams. Clusters could be easily distinguished containing only words such as "highly recommend" or "waste money". For the DBPedia dataset, this method however found different clusters when increasing normalization, for example a cluster on academic journals or one on science fiction. This is also reflected in the decreasing ARI and AMI in Figure 2. For DBPedia, TFIDF was more intuitive. However one cannot say if the highest level in TFIDF would be better than a lower level in TFIDF N-gram by only looking at the terms.

5 Discussion

The anticipated result was to observe a general effect of normalizing text that would affect the evaluation metrics such that there is a certain point where normalization complexity has reached its maximum. This result was not observed in all results and this study can, therefore, not conclude that there is a generalized norm for preprocessing short text documents for cluster analysis. However, the anticipated result can be observed in some cases together with other connections between text cleaning and feature extraction. The results for Yelp and Amazon are often similar which could be a result of both datasets being similar. It is also good to indicate that DBPedia has results that are often of a higher magnitude than the other datasets, but still the effects of preprocessing are identical which give the observation more strength.

Only GloVe and TFIDF N-gram are able to get similar results across all datasets. GloVe shows almost identical results for every dataset and metric. This does however not mean that the same normalization levels work for all datasets, it means that there is a consistency between the metrics per dataset. Overall, GloVe prefers level 1 for Yelp and level 3 for Amazon and DBPedia for all metrics. This implies that in order to maximize recovery of the true cluster membership while maximizing similarity within clusters, GloVe can be used. This consistency is not observed in other feature extraction methods, and for these one has to make clear what the goal of clustering is, either maximizing the recovery of true clustering or similarity within clusters, since the ASW and ARI are often opposites. Looking at the top terms per cluster, the discussed results for GloVe are only in line for the DBPedia dataset, where the clusters can be distinguished. In TFIDF N-gram the best normalization level to maximize ASW is level 3 for all datasets. This observation indicates that ASW is dataset independent for TFIDF N-gram. Thus it is expected that for any dataset when using TFIDF N-gram the ASW will be best when text is in lowercase and has punctuation, rare words, common words and stopwords removed. This is again in line with looking at the top terms per cluster, where the clusters become distinguishable from level 3 onwards. Further, TFIDF N-gram with lemmatization applied has a smaller dimensionality, but comes close to the best results and could therefore be a good alternative.

In addition to these results, for most graphs a decrease is observed when stemming is applied and an increase once lemmatization is applied, although these might not be the best level in general. When looking at the top terms per cluster, these two levels seem equal only without endings when stemming is applied. Intuitively lemmatization works better than stemming on short text documents because it is able to capture more context. Stemming may create more redundancy by introducing words that are not recognized by the stopword remover or at all. Overall, for the datasets used in this study the best method was Word2Vec for maximizing ASW. Since Word2Vec was designed such that it was trained on the text itself, it was able to focus on the main message in every document and increase document similarity within clusters. Interpreting the top terms for Word2Vec was however difficult since no clear trend could be observed along the normalization levels.

6 Conclusion

In this paper, the effect of preprocessing and feature extraction was analyzed on short documents. Six levels of preprocessing with increasing complexity were used together with four different feature extraction methods. All combinations were then tested with K-means clustering using three different datasets. All details of the methods used were discussed together with justifications. The results of this study did not show the anticipated result of having a maximum normalization complexity which is generalized for all models. However, other insights were discovered. An important result being that GloVe can be used for feature extraction in research that wants both to increase within cluster similarity and recovery of the true clusterings. Another main result is that TFIDF with N-grams is found to be dataset independent for within cluster similarity (ASW). Further, lemmatization is observed to be better than stemming and Word2Vec works best for the datasets used. Future work would include testing the same study on different cluster algorithms such that better generalization can be achieved. Comparing more feature extraction methods and more levels of normalization, could also be investigated and may lead to clearer results. Other research could include analyzing different sizes of datasets and also varying the number of clusters per dataset. In general there are many aspects that can be investigated within clustering short text documents, this study contributes to this research field and found relationships between feature extraction and preprocessing that were not expected.

References

Adel A, Elfakharany E, Badr A (2014) Clustering tweets using cellular genetic algorithm. Journal of Computer Science 10(7):1269–1280. DOI: 10.3844/jcssp.2014.1269.1280.

Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K (2017) A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. arXiv:1707.02919. URL: `http://arxiv.org/abs/1707.02919`.

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. Journal of Machine Learning Research 3:993–1022. URL: `http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf`.

Crockett K, Mclean D, Latham A, Alnajran N (2017) Cluster Analysis of Twitter Data: A Review of Algorithms. In: Proceedings of the 9th International Conference on Agents and Artificial Intelligence, van den Herik J, Rocha AP, Filipe J (eds), INSTICC, SciTePress, Vol. 2, pp. 239–249. ISBN: 978-9-897582-20-2, DOI: 10.5220/0006202802390249.

Damashek M (1995) Gauging similarity with n-grams: Language-independent categorization of text. Science 267(5199):843–848, American Association for the Advancement of Science. DOI: 10.1126/science.267.5199.843.

Ellen J (2011) All about Microtext - A Working Definition and a Survey of Current Microtext Research within Artificial Intelligence and Natural Language Processing. ICAART 1:329–336. URL: `https://www.scitepress.org/Papers/2011/31799/31799.pdf`.

Fan W, Gordon MD (2014) The Power of Social Media Analytics. Communications of the Association for Computing Machinery 57(6):74–81. DOI: 10.1145/2602574.

Fellbaum C (1998) WordNet: An Electronic Lexical Database. MIT press.

Hubert L, Arabie P (1985) Comparing partitions. Journal of classification 2(1):193–218, Springer. DOI: 10.1007/BF01908075.

Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, Hellmann S, Morsey M, Van Kleef P, Auer S, et al (2015) DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web 6(2):167–195, IOS Press. DOI: 10.3233/SW-140134.

Leskovec J, Krevl A (2014) SNAP Datasets: Stanford Large Network Dataset Collection. URL: `http://snap.stanford.edu/data`.

Liu L, Kang J, Yu J, Wang Z (2005) A comparative study on unsupervised feature selection methods for text clustering. In: Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on, IEEE, Wuhan, pp. 597–601. DOI: 10.1109/NLPKE.2005.1598807.

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed Representations of Words and Phrases and Their Compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ (eds), Curran Associates Inc., Red Hook, NY, USA, NIPS'13, pp. 3111–3119. DOI: 10.5555/2999792.2999959.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12:2825–2830. URL: `http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf`.

Pennington J, Socher R, Manning CD (2014) GloVe: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Moschitti A, Pang B, Daelemans W (eds), Association for Computational Linguistics, Doha, pp. 1532–1543. URL: `https://www.aclweb.org/anthology/D14-1162/`.

Poomagal S, Visalakshi P, Hamsapriya T (2015) A novel method for clustering tweets in twitter. International Journal of Web Based Communities 11(2):170–187. DOI: 10.1504/IJWBC.2015.068540.

Porter MF (1980) An algorithm for suffix stripping. Program 14(3):130–137. DOI: 10.1108/00330330610681286.

Řehůřek R, Sojka P (2010) Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Calzolari N, Choukri K, Maegaard B, Mariani J, Odijk J, Piperidis S, Rosner M, Tapias D, Mazo H, Hamon O, Russo. I (eds), European Language Resources Association, Valletta, pp. 45–50. DOI: 10.13140/2.1.2393.1847.

Robertson S (2004) Understanding inverse document frequency: on theoretical arguments for IDF. Journal of documentation 60(5):503–520. DOI: 10.1108/00220410410560582.

Romano S, Vinh NX, Bailey J, Verspoor K (2016) Adjusting for chance clustering comparison measures. The Journal of Machine Learning Research 17(1):4635–4666, Nowozin S (ed).

Rosa KD, Shah R, Lin B, Gershman A, Frederking R (2011) Topical clustering of tweets. Proceedings of the ACM SIGIR: SWSM. URL: `http://www.cs.cmu.edu/~kdelaros/sigir-swsm-2011.pdf`.

Rousseeuw PJ (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20:53–65. DOI: 10.1016/0377-0427(87)90125-7.

Tang B, Shepherd M, Milios E, Heywood MI (2005) Comparing and combining dimension reduction techniques for efficient text clustering. In: Proceeding of SIAM international workshop on feature selection for data mining, pp. 17–26. URL: `https://web.cs.dal.ca/~eem/cvWeb/pubs/btang-FSDM-2005.pdf`.

Uysal AK, Gunal S (2014) The impact of preprocessing on text classification. Information Processing & Management 50(1):104–112. DOI: 10.1016/j.ipm.2013.08.006.

Vinh NX, Epps J, Bailey J (2010) Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research 11:2837–2854. URL: `http://www.jmlr.org/papers/v11/vinh10a.html`.

Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. In: Advances in neural information processing systems, Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds), Curran Associates, Inc., pp. 649–657. URL: `https://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification`.