Information Processing and Management 57 (2020) 102034

Contents lists available at ScienceDirect

Information Processing and Management

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

An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit

Stephan A. Curiskis*, Barry Drake, Thomas R. Osborn, Paul J. Kennedy

Centre for Artificial Intelligence Faculty of Engineering and Information Technology University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007 Australia

ARTICLE INFO

Keywords:
Document clustering
Topic modelling
Topic discovery
Embedding models
Online social networks

ABSTRACT

Methods for document clustering and topic modelling in online social networks (OSNs) offer a means of categorising, annotating and making sense of large volumes of user generated content. Many techniques have been developed over the years, ranging from text mining and clustering methods to latent topic models and neural embedding approaches. However, many of these methods deliver poor results when applied to OSN data as such text is notoriously short and noisy, and often results are not comparable across studies. In this study we evaluate several techniques for document clustering and topic modelling on three datasets from Twitter and Reddit. We benchmark four different feature representations derived from term-frequency inverse-document-frequency (*tf-idf*) matrices and word embedding models combined with four clustering methods, and we include a Latent Dirichlet Allocation topic model for comparison. Several different evaluation measures are used in the literature, so we provide a discussion and recommendation for the most appropriate extrinsic measures for this task. We also demonstrate the performance of the methods over data sets with different document lengths. Our results show that clustering techniques applied to neural embedding feature representations delivered the best performance over all data sets using appropriate extrinsic evaluation measures. We also demonstrate a method for interpreting the clusters with a top-words based approach using *tf-idf* weights combined with embedding distance measures.

1. Introduction

In January 2018 there were estimated to be around 4.021 billion people around the world who use the internet. Of these, 3.196 billion people use social media in some form, generating a staggering amount of content.[1] Online platforms and social networks have become a key source of information for nearly half of the world's population. These platforms are increasingly being used to disseminate information regarding news, brands, political discussion, global events and more (Bakshy, Rosenn, Marlow, & Adamic, 2012). However, much of the data generated is unstructured and not annotated. This means that it is difficult to understand how topics of information are diffused through online social networks (OSNs), and how users engage with different topics (Guille, Hacid, Favre, & Zighed, 2013). Automatically annotating topics within OSNs may facilitate analysis of information diffusion and user preferences by enriching the data available from these platforms, in a way that is readily analysed. With the rise of phenomena like echo chambers and filter bubbles, which lead to individuals receiving biased and narrowly focused content, the challenge of

* Corresponding author.
E-mail address: stephan.a.curiskis@student.uts.edu.au (S.A. Curiskis).

[1] https://wearesocial.com/uk/blog/2018/01/global-digital-report-2018, accessed Sep. 2018.

https://doi.org/10.1016/j.ipm.2019.04.002
Received 22 September 2018; Received in revised form 11 April 2019; Accepted 11 April 2019
Available online 17 April 2019
0306-4573/

automatically annotating OSN data has become important.

Document clustering is a set of machine learning techniques that aim to automatically organise documents into clusters such that documents within clusters are similar when compared to documents in other clusters. Many methods for clustering documents have been proposed (Bisht & Paul, 2013; Naik, Prajapati, & Dabhi, 2015). These techniques typically involve the use of a feature matrix, such as a term-frequency inverse-document-frequency matrix (*tf-idf* matrix) to represent a corpus, with a clustering method applied to this matrix. More recently, representations derived from neural word embeddings have seen applications on social media data as they can produce dense representations with semantic properties and require less manual preprocessing than traditional methods (Li, Shah, Liu, & Nourbakhsh, 2017). Common clustering methods applied in this context build hierarchies or partitions (Irfan et al., 2015). Example hierarchical methods are agglomerative clustering and divisive clustering. Example partitioning methods are k-means and k-medoids clustering.

Topic modelling involves methods to discover patterns of word use within documents, and is an active research area with several techniques recently applied to OSN data (Chinnov, Kerschke, Meske, Stieglitz, & Trautmann, 2015). Topics are typically defined as a distribution of words, with documents modelled as mixtures of topics. Like document clustering, topic modelling can be used to cluster documents by giving a probability distribution over a range of topics for each document. This can be viewed as a form of soft partition clustering, where the data points have a probabilistic degree of ownership to each cluster. The topic representation also provides the word distribution for each topic which aids in interpretation. Commonly used topic models with applications on OSN text data include Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2003), the Author-Topic model (Hong & Davison, 2010), and more recently Dynamic Topic Models which discover topics over time (Alghamdi & Alfalqi, 2015).

Document clustering and topic modelling are increasingly important research areas as these methods can be applied to large amounts of readily available OSN text data, yielding homogeneous groups of documents. These document groups may then align to relevant topics and trends. Clustering is particularly suited to OSN data as platforms like Twitter and Facebook use hashtags as a form of topic annotation (Steinskog, Therkelsen, & Gambäck, 2017), which may be used for evaluation of document clustering and topic modelling methods. Large scale clustering can help make sense of the huge amount of content being created online every day, and can subsequently be used in further machine learning tasks. Additional features derived from OSN data (such as user demographic, geographic and network data) have also been clustered to find groups of online posts or comments that are semantically similar (Alnajran, Crockett, McLean, & Latham, 2017). However, OSN data presents many challenges when applying topic modelling and document clustering methods. For example, such text is typically short and contains noise such as misspellings and grammatical errors (Chinnov et al., 2015).

There are two key challenges with topic modelling and document clustering research on OSN data sets. Firstly, results are often not reproducible since the data used in the studies frequently cannot be published. For instance, Twitter's terms of service do not allow for tweets to be published. Instead, researchers can publish a list of tweet identifiers that were used and retrieved via the API. Unfortunately, over time the associated tweets are removed from the platform, which degrades the underlying data. The data sets used are also often small or biased towards particular contexts. These issues result from the complex data collection and preparation that is often required to extract large data sets from an OSN platform, as well as restrictions on the platforms themselves (Stieglitz, Mirbabaie, Ross, & Neuberger, 2018).

Secondly, different studies often use different methods for evaluating the performance of clustered documents. Evaluation methods on Twitter data vary from extrinsic measures which compare clusters against labelled data, to manual assessments of cluster performance and interpretability (Alnajran et al., 2017). It is therefore difficult to compare empirical results. With the fast pace of research in this area, there is little guidance on what method or family of methods will perform best in specific circumstances, such as on short Twitter data or relatively longer Reddit comments.

In this paper we provide an analysis of the performance of several methods for document clustering and topic modelling of OSN content on three data sets: two Twitter data sets and a publicly available Reddit data set. We evaluate four feature representation methods derived from *tf-idf* and embedding matrices combined with four clustering techniques, and include a Latent Dirichlet Allocation (LDA) topic model for comparison. We also provide a discussion of the properties and appropriateness of document clustering evaluation measures commonly used in the literature. We evaluate performance with three such measures, namely the Normalised Mutual Information (NMI), the Adjusted Mutual Information (AMI), and the Adjusted Rand Index (ARI). Furthermore, we have made our data sets available so that our results can be reproduced. To comply with Twitter's terms of use, we have made available the tweet identifiers used along with the topic label. We have also made available the full Reddit data set used (Curiskis, Drake, Osborn, & Kennedy, submitted).

Further to this, by tuning key hyper-parameters we demonstrate how embedding models can be used to generate feature sets for document clustering that delivered good performance and captured latent structure in the data. We also show how word embedding distances can aid in the interpretation of the clusters by ranking the top words, forming a topic vector of words. This contribution is significant since data sets from OSNs are often short and contain noise such as misspellings, abbreviations, acronyms, special characters, emojis, URLs and hashtags. These issues can result in poor performance for many commonly used techniques. Furthermore, a clear consensus is lacking in the literature regarding methods that work effectively on OSN data. The results of this paper provide guidance on methods giving good performance over different types of OSN data. These results show that traditional topic modelling and document clustering approaches do not work well on short and noisy social media posts. Instead, clustering approaches applied to more recent neural network embedding representations can deliver improved performance.

The structure of this paper is as follows. In Section 2 we review the current literature in this research area. In Section 3 we present the detail of our methods, including a description of the data extraction, the preparation process, the feature representations, the clustering methods, and the evaluation measures. In Section 4 we present our results with a discussion. In Section 5 we provide a

discussion followed by our conclusion in Section 6.

2. Literature review

We organize the literature on document clustering and topic modelling of OSNs into three areas. Firstly, many studies have centred on identifying and interpreting memes in this domain, incorporating textual, network and user data. Secondly, identifying topics through topic models and clustering approaches has received much attention as a means of understanding and categorising online content. Thirdly, recent advances in neural word embedding models have been used to provide dense feature representations of documents from OSNs.

2.1. Meme identification

The term "meme" is commonly used to represent an element of culture or system of behaviour that spreads from one individual to another by imitation. In the context of OSNs, for this paper we define a "meme" as a semantic unit expressed as electronic text where the semantics are transferred across multiple individuals even though the text may be different. This specific definition of "meme" is sometimes called "ememe" (Shabunina & Pasi, 2018). A topic in OSN applications can be defined as a coherent set of semantically related terms which express a single argument (Guille et al., 2013). In comparison to this definition of a topic, a meme does not necessarily need to be derived from a set or distribution of words, but instead aims to detect significant semantic content. Often in practice, however, there is an overlap between the two concepts. The concept of a meme is useful for OSN applications as it can be thought of as a latent representation of textual content, but can also be discovered through analysis of OSN user and network data.

A study by Ferrara et al. (2013) aimed to identify memes within large social media data. In that study, several similarity measures were defined for Twitter data which leverage content, metadata and network features. The authors defined the concept of a 'protomeme' which was used to refer to hashtags, user mentions, URLs and phrases. Data was aggregated by creating protomeme projections onto spaces based on tweet, user and content features. For each protomeme pair, common user, tweet, content and diffusion similarity measures were calculated. These similarity matrices were then aggregated in several different ways, such as the element-wise mean and maximum. Finally, the aggregated similarity matrix was clustered with hierarchical clustering. The resulting clusters were taken to represent memes within the data. The data set used was a collection of 5523 tweets related to the US presidential primaries in April 2012. Twenty-six topics were manually identified and assigned as labels to each tweet. Since the memes and topics can overlap per tweet, performance was evaluated using a variation of Normalised Mutual Information designated as LFK-NMI. Given the optimal parameters for this approach, the protomeme clustering method delivered average 5-fold cross-validation LFK-NMI scores of around 0.13. JafariAsbagh, Ferrara, Varol, Menczer, and Flammini (2014) later extended the algorithm to work on streaming data.

More recently, Shabunina and Pasi (2018) developed a method to identify and characterise memes, considered as a set of frequently occurring related words propagating through a network over time. The relationships between terms in a social media stream were modelled using a graph of words. To identify memes, a *k*-core degeneracy process was applied to the graph to generate subgraphs, which constituted meme bases. A meme was defined as the fuzzy subset of terms in a meme basis. The method was applied to over 800,000 tweets from the search queries #economy, #politics and #finance. Although useful to characterise and interpret topics in social media streams, memes were not attributed to individual social media documents or users. Evaluation of the method was limited to subjective interpretation and intrinsic measures.

2.2. Document clustering and topic modelling

In contrast to methods for meme identification, many studies have focused on detecting topics in OSNs. Topic models typically refer to methods that group both documents, which share similar words, as well as words that occur in a similar set of documents. Document clustering refers to methods that group documents according to some feature matrix, such that documents within a cluster are more similar to documents in other clusters. Due to the short document size and high degree of noise inherent OSN data, such as Twitter data, clustering based methods are often applied in favour of more traditional topic models (Chinnov et al., 2015). Nevertheless, topic models applied to OSN data are still an active area of research (Alghamdi & Alfalqi, 2015). Indeed, the term 'topic discovery' may refer to either topic modelling or document clustering.

Document clustering methods have typically used vector space representations of word occurrence by document. Commonly, bag-of-words methods model each document as a point in the space of words. Each word is a feature or dimension of this space, with element values assigned in one of several ways. These can be one-hot-encodings, where the value is set to 1 if the word exists in the document and 0 otherwise, term frequency, or term-frequency inverse-document-frequency calculations. Given that the total dimension size is the number of unique words, often there is a threshold cut-off to use only those words with high values (Patki & Khot, 2017). A range of clustering algorithms may then be applied to the feature matrix, such as k-means, hierarchical clustering, self-organising maps, and so on (Naik et al., 2015).

For instance, Godfrey, Johns, Meyer, Race, and Sadek (2014) developed an algorithm to identify topics within a specific Twitter data set, a collection of about 30,000 tweets extracted using the query term 'world cup'. Non-negative Matrix Factorisation (NMF) and k-means clustering were applied to the *tf-idf* representation of tweets to create topic clusters. Due to the noisiness of Twitter data, Godfrey et al. (2014) developed a preliminary filtering step using multiple runs of the DBSCAN clustering algorithm combined with consensus clustering. The rationale was that tweets which are not close to any particular cluster may be treated as noise and removed from an analysis. The results when using this approach showed that both k-means clustering and NMF produced similar results.

However, when analysing the clusters using a subjective evaluation of tweet network diagrams and word clouds, NMF seemed to produce more interpretable clusters.

Fang, Zhang, Ye, and Li (2014) approached detecting topics in Twitter using additional information about the tweet. Recognising that the textual content of tweets can be quite limited, a 'multi-view' topic detection framework was developed based on more granular 'multi-relations'. These multi-relations were defined as useful relations from the Twitter social network and included hashtags, user mentions, retweets, meaningful words and similar posting times. To measure these multi-relations, a document similarity measure was developed. Multi-relation similarity scores were then combined into a multi-view and clustered using three different methods. These clusters were taken to represent topics and a keyword extraction method, based on suffix trees and *tf-idf* weights, was applied to derive representative keywords for each cluster. This method was evaluated using a dataset of 12,000 tweets with 60 'hot' topics extracted from the Twitter API. Three evaluation measures were used, namely the *F*-measure, NMI, and entropy. The results showed that including more multi-views improved performance, with results above 0.928 on the *F*-measure and 0.935 NMI. However, the authors did not remove any of the hot topic key words from the text. These key words are generally short phrases or hashtags, and can be discovered easily by *tf-idf* approaches.

Another study compared the efficacy of different clustering methods to detect topics in Twitter data centered around recent earthquakes in Nepal (Klinczak & Kaestner, 2016). In this study, tweets were represented by their *tf-idf* vectors. Four clustering methods applied to this representation were compared, namely k-means, k-medoids, DBSCAN and NMF. By evaluating each clustering method with measures for cohesion and separation of clusters (i.e. intrinsic evaluation measures), it was clear that NMF produced superior clusters which were simpler and easier to interpret. More recently, Suri and Roy (2017) applied LDA and NMF to detect topics on a Twitter data set, as well as a RSS news feed. Both methods were found to have similar performance. LDA was deemed to be more interpretable, but NMF was faster to calculate. However, performance was evaluated by manual inspection of the key terms for topics.

Many studies have applied topic modelling techniques to OSN data. For instance, Paul and Dredze (2014) developed a topic modelling framework for discovering self-reported health topics using Twitter data. 5128 tweets were annotated with a positive status if they related to the user's health, and negative if not. A logistic regression model was trained to predict the positive labels in the annotated data, and applied to a Twitter stream filtered with a large number of health related keywords. This provided a set of 144 million health tweets which was used to run the Ailment Topic Aspect Model. While this study is useful in filtering and interpreting large amounts of relevant tweets, validation of the discovered topics focused on correlation measures against external health trend data.

Further to topic models applied to a static data set, dynamic topic models, which incorporate the temporal nature of OSN data, are gaining attention (Alghamdi & Alfalqi, 2015). Ha, Beijnon, Kim, Lee, and Kim (2017) applied dynamic topic models to Reddit data to understand user perceptions of smart watches. While these results are interesting to gauge public opinion in this area, no ground truth label was used and likewise no extrinsic evaluation measures were applied. Recently, Klein, Clutton, and Polito (2018) applied topic modelling to reveal distinct interests in the Reddit conspiracy page (a subreddit page). NMF was used to create topic loadings for each user contributing to the page. These topic loadings were then clustered using k-means to reveal user subgroups. Again, this study is useful in understanding the user population within OSN discussion threads, but no extrinsic evaluation was made to validate the quality of the topic modelling or the clustering.

2.3. Neural network embedding models

Much of the literature on clustering OSN text data used *tf-idf* matrix representations of tweets at some level. These matrices treat terms as one-hot encoded vectors, where each term is represented by a binary vector with exactly one non-zero element. This means that relationships between words, such as synonyms, are not incorporated and the resulting document matrix representation is sparse and high dimensional. The concept of dense, distributional representations of words, or word embeddings, provide an alternative approach (Bengio, Ducharme, Vincent, & Janvin, 2003). In these methods, each word is represented by a real valued vector of fixed dimension. Word embeddings are commonly trained using neural network language models, such as word2vec (Mikolov, Chen, Corrado, & Dean, 2013). However, when using word embedding models to create document level representations, the word vectors need to be aggregated in some way. Common approaches in the literature are to simply take the mean of the word vectors for all terms in the document, or to concatenate the vectors to a document vector of fixed size (Yang, Macdonald, & Ounis, 2017). Document representations derived from *tf-idf* weighted word vector averages have also been proposed (Corrêa Júnior, Marinho, & dos Santos, 2017; Zhao, Lan, & Tian, 2015). Another method trains document level dense vector representations at the same time as the word vectors (Le & Mikolov, 2014). We refer to this latter method as doc2vec.

Much research has applied neural word embeddings to classification and semantic evaluation tasks. For instance, Billah Nagoudi, Ferrero, and Schwab (2017) applied word embeddings to model semantic similarity between Arabic sentences. Three different sentence level aggregations were proposed, namely the sum of the word vectors for all words in a sentence, an inverse-document-frequency weighted sum of the word vectors, and a part-of-speech weighted sum. The authors found that the weighted sum representations delivered more accurate sentence similarities. In another study, Corrêa Júnior et al. (2017) developed a classification method for sentiment analysis using an ensemble of classifiers with different feature representations, namely a *tf-idf* matrix, a mean word vector representation, and a *tf-idf* weighted mean of the word vectors. Recently, Li et al. (2017) published a number of pre-trained word2vec models on a Twitter data set of 390 million English tweets with a range of pre-processing steps. Embedding representations are becoming more widely used in NLP tasks involving OSN data.

Further to word and document embeddings, character level embedding models have been proposed and applied to Twitter data,

creating *tweet2vec* (Dhingra, Zhou, Fitzpatrick, Muehl, & Cohen, 2016). The motivation for *tweet2vec* is that social media data are noisy, suffering from spelling errors, abbreviations, acronyms and special characters, which can lead to prohibitively large vocabulary sizes. *Tweet2vec* takes as input sequences of characters for each tweet and passes them through a bidirectional GRU neural network encoder to create a fixed dimensional tweet embedding vector. This tweet embedding is then passed through a linear softmax layer to predict the hashtags of a tweet. The algorithm was evaluated on hashtag classification performance. While this method may promise to create useful tweet embeddings, it assumes that hashtags are valid labels for tweets. This assumption may not hold as other text, user mentions and URLs can also be important in defining the topic of the tweet, and tweets can have multiple hashtags.

Recently, contextualised extensions to word embeddings have been proposed. One challenge for traditional word embeddings is polysemy, where a word has multiple meanings dependent on the context. Peters et al. (2018) introduced a deep contextualised word embedding model, which models both the syntactic and semantic characteristics of word use, and how these uses vary across linguistic contexts. This method involves coupling embedding vectors trained from a bidirectional LSTM with a language model objective. Named *ELMo* (Embeddings from Language Models), the method assigns an embedding vector to each token that is a function of the entire input sentence. This technique may be useful for clustering social media documents.

In addition to the document clustering and topic modelling approaches discussed so far, a new series of deep learning based clustering methods have been developed (Min et al., 2018). Many of these techniques use deep neural networks to learn feature representations trained at the same time as clustering. Examples include several deep autoencoder networks with a clustering layer, where the loss function is a combination of reconstruction loss and clustering loss. Clustering methods based on generative models such as Variational Autoencoders and Generative Adversarial Networks look promising from a document clustering perspective since they can also generate representative samples from the clusters. However, the focus for these techniques to date has been on image data sets.

Many approaches to document clustering and topic modelling are proposed for OSN text data. These methods typically involve creating document level feature representations with *tf-idf* matrices or other techniques, followed by clustering methods to group documents into semantically related clusters. However, there are many variations on these methods and word embedding representations have not yet been effectively applied and benchmarked on document clustering tasks in OSN data, to the best of our knowledge.

3. Methods

In this section we describe the three data sets used and the processing steps, the feature representations and clustering algorithms, and the evaluation measures used with a discussion of their properties.

Document clustering and topic modelling methods applied to OSN data typically involve several processing steps as outlined in Fig. 1. Data is first extracted from a source. From the raw data set or OSN platform API, documents are extracted which consist of text data from an individual user. A tweet and a Reddit parent comment are examples of a document. The textual elements are then processed to remove common punctuation and stop words, and tokenised. Feature representations of each document are created,

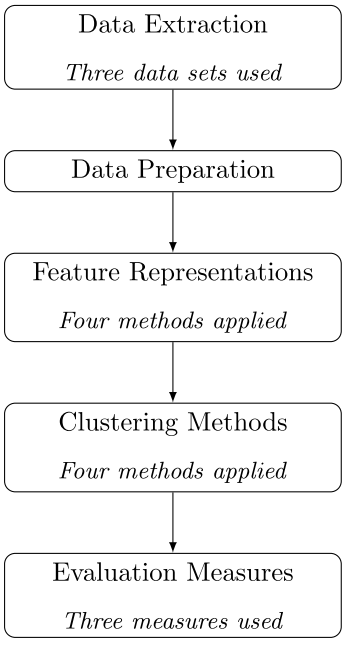


Fig. 1. Process pipeline for document clustering. The contribution of this paper is an evaluation of four methods for feature representation and four clustering methods using three evaluation measures over three data sets.

Table 1
Outline of the data sets, methods for feature representations and clustering, and extrinsic evaluation measures used in this study. For the three data sets, we evaluate the feature representation and clustering method combinations and the LDA topic model (17 combinations) with the three evaluation measures.

Data sets
Twitter stream filtered by #Auspol, 29,283 tweets
RepLab 2013 competition Twitter data, 2657 tweets
Reddit data from May 2015, 40,000 parent comments
Methods
Feature representations:
FR1 *tf-idf* matrix with the top 1000 terms per document
FR2 Mean word2vec matrix
FR3 Mean word2vec matrix weighted by the top 1000 *tf-idf* scores
FR4 doc2vec matrix for each document
Clustering methods:
CM1 k-means clustering
CM2 k-medoids clustering
CM3 Hierarchical agglomerative clustering
CM4 Non-negative matrix factorisation (NMF)
Topic model:
LDA Latent Dirichlet Allocation topic model
Evaluation Measures
NMI Normalised Mutual Information
AMI Adjusted Mutual Information
ARI Adjusted Rand Index

followed by a clustering method. Extrinsic clustering evaluation measures are then calculated using ground truth labels. The variations at each step of the process are outlined in Table 1. In the rest of this section we detail our approach to each step of Fig. 1.

3.1. Data extraction

We used three OSN data sets for evaluation; two Twitter data sets and a Reddit data set. We have used Twitter data since it has been widely used in the literature regarding topic modelling and document clustering. While there appear to be fewer studies which have used Reddit data, Reddit still represents a valuable source of OSN data to use for topic modelling and document clustering. Reddit is also used more as a discussion forum, and comments have a wider range of document lengths than Twitter data. All three data sets have been made available (Curiskis et al., submitted).

Twitter data provides a readily accessible data source for short and topical user driven content. It is also widely used for research purposes, but has many challenges due to the short tweet length and use of hashtags, acronyms, user mentions and URLs (Stieglitz et al., 2018). The first Twitter data set was collected through Twitter's public API. It was constructed by filtering the Twitter stream for the hashtag #Auspol, which is frequently used in Australia for political discussion. A common application for document clustering on OSN data is to take a set of documents related to a particular theme and discover topics, such as the study of health topics in Twitter data (Paul & Dredze, 2014). The #Auspol Twitter data set is suitable for comparing document clustering methods since the hashtag is widely used to link a large number of disparate discussions, often with additional hashtags, related to public opinion in Australia. Data was collected between 13 June and 2 September 2017 and consisted of 1,364,326 tweets. We filtered this data set by selecting English language tweets only and removed retweets based on the *retweeted_status* field and a text filter. This resulted in 205,895 tweets.

No ground truth topic labels exist for this data set so we used a set of high count hashtags as ground truth labels. We further removed the search hashtag (#Auspol) from the data set, since all tweets had this token. It is common for there to be multiple hashtags on a tweet, so to avoid having overlapping topics we removed tweets which contained more than one of the top hashtags. We also manually removed some related hashtags, such as #ssm (same sex marriage) which is closely related to #marriageequality; we kept the latter as it was used in more tweets. Lastly, we filtered by hashtags with at least 1000 tweets to keep the topics relatively balanced. This resulted in 29,283 tweets with 13 hashtags denoting topic labels, as given in Table 2.

The second Twitter data set was taken from the RepLab 2013 competition (Amigó et al., 2013). This competition focused on monitoring the reputation of entities (companies and individuals), and involved tasks such as named entity recognition, polarity classification and topic detection. The tweets used in this competition were annotated with topic labels by several trained annotators supervised and monitored by reputation experts. For the purposes of this paper, the topics annotated in these tweets were taken as a gold standard. We have used this data set because it has gold standard labels already annotated and has been used for topic detection tasks.

We downloaded the list of Twitter identifiers from the training and testing data sets for the topic detection task made available through the RepLab 2013 competition and retrieved the details through the Twitter API on 19 January, 2019. Out of 110,344 published tweet identifiers with labelled topics, we could only retrieve the tweet text and other information for 23,684 tweets. This is

Table 2
Count of tweets per hashtag in the #Auspol Twitter data set.

Topic number	Hashtag	Tweets
1	#qldpol	3845
2	#qanda	3592
3	#insiders	3495
4	#lnp	3434
5	#politas	2618
6	#marriageequality	2562
7	#springst	1708
8	#nbn	1626
9	#trump	1547
10	#uspoli	1498
11	#stopadani	1186
12	#climatechange	1148
13	#turnbull	1024

likely due to tweets and users being deleted since the tweets were published. Furthermore, there is a long tail of topics labelled in this data. In fact, for the 23,684 tweets there were a total of 3432 distinct topics, with 1263 topics containing a single tweet. To ensure that there were sufficient data points for our methods to detect, we limited the frequency count per topic to be 100. We also removed the label denoted 'other topics' as this does not represent an internally consistent topic. After this filtering we had a data set of 2657 tweets with 13 topic labels from the competition. The list of topic labels used is given in Table 3.

We originally included the RepLab 2013 data set primarily because comparative results for topic discovery are available from the competition. However, due to the large volume of tweets which could not be retrieved from Twitter's API, accurate comparisons are no longer possible. Nevertheless, the ground truth topic labels still allow for the performance of the methods to be benchmarked.

The third data set was from the Reddit platform and consisted of parent comments and their related comments by Reddit subreddit page from May 2015. The Reddit platform is widely used for discussion related to specific topics or themes, grouped by subreddit page, so is ideal for this study. Furthermore, Reddit comments can be longer than tweets. Reddit parent comments refer to the top comment which may or may not have responses from other users. This data was made public on the Reddit website (Reddit, 2015). The full data set contained around 54.5 million comments on 50,138 subreddit pages. We chose this data set since it is freely available in full and contains discussion on multiple themes. It is therefore an ideal data set to use for benchmarking methods. We chose five subreddit pages which represent disjoint themes for analysis. These five subreddit pages were also used in a previous study benchmarking classification models (Gutman & Nam, 2015). Since parent comments and responses are inherently related, we pooled all the user posts into documents grouped by the parent comment identifier. Table 4 shows the count of parent comments per subreddit page. We randomly sampled 40,000 parent comment identifiers from across the five subreddit pages, then used these pages to denote the ground truth labels.

Reddit data is especially useful in this study since it contains a wider range of character lengths per document than Twitter data, since Twitter has a limit on the number of characters. An evaluation of the performance of the document clustering methods by document length can provide guidance for future studies on the optimal method for a particular data set. To examine this performance, we partitioned the Reddit data into four distinct subsets based on the number of characters per document. Details for the four data partitions are given in Table 5. For comparison with the Twitter data sets, a tweet has a maximum of 240 characters. For the #Auspol Twitter data, the mean character length was 117 with 25th percentile of 103 and 75th percentile of 138. Most tweets therefore fall into the 101 to 200 character length document group.

Table 3
Count of tweets per topic label in the RepLab 2013 Twitter data set.

Topic number	Topic	Tweets
1	For sale	329
2	Suzuki cup	296
3	User comments	262
4	Money laundering / terrorism finance	199
5	Record of views on YouTube	195
6	Fan Craze - Beliebers	154
7	Princeton offense	131
8	For Sale - Nissan Cars, Parts analysed Accessories	127
9	Jokes	127
10	Sports sponsors	127
11	Spam	114
12	Ironic criticism	111
13	MotoGP - User comments	103

Table 4
Count of parent comments per subreddit page.

Topic number	Subreddit page	Parent comments
1	NFL	10,563
2	News	9488
3	pcmasterrace	9186
4	Movies	6263
5	Relationships	4500

Table 5
Reddit data was partitioned into four sets based on document character length. Documents are grouped by the parent comment. The mean character length and mean number of tokens per document are given.

Character length range	Number of documents	Mean character length	Mean number of tokens
1–100	15,273	46.1	4.5
101–200	8360	144.9	13.3
201–500	9310	317.4	28.6
501 or greater	7057	1,584.5	141.1

3.2. Data preparation

Data preparation and analysis in this study was conducted using *python 3.6.1*. For text preprocessing, we removed the list of stopwords from the *nltk 3.2.4* package and punctuation from *string*. A customised tokeniser function was created for tweets which retained hashtags and user mentions, and removed URLs. To tokenise the Reddit data, we simply removed punctuation and standard stopwords. We did not apply any stemming or lemmatisation. We also used the *TfidfVectorizer* function from *sklearn 0.19.1* for the *tf-idf* method and the weighted word2vec method.

For the #Auspol Twitter data, we removed the list of 14 hashtags taken as ground truth labels from the text, in addition to the #Auspol Twitter API search query. The RepLab 2013 Twitter data set had annotated topic labels that were not based directly on any individual tokens, so no modification was required. For the Reddit data, as the subreddit page was used as the ground truth label we did not need to modify the text.

3.3. Feature representations

In this study we evaluated the performance of four methods to construct feature representations for documents combined with four commonly used clustering algorithms. We also included an LDA topic model in a separate topic models category since the technique only takes as input a bag-of-words matrix. These methods are outlined in Table 1, where each method component is given a code for ease of reference. The four feature representations are coded as **FR1-FR4** and the four clustering methods are coded as **CM1-CM4** and the LDA topic model is coded simply as **LDA**. While many other techniques have been proposed in the literature, such as the meme identification studies (JafariAsbagh et al., 2014; Shabunina & Pasi, 2018), we did not implement them for evaluation as they are specific to data from Twitter. However, we provide comparison results in our discussion where they were available from other studies.

For **FR1**, the *tf-idf* matrix was limited to the top 1000 terms per document by frequency since no performance improvement was gained by including more terms. This is likely due to the short nature of social media text which produces sparse *tf-idf* feature vectors; terms with lower frequency would not generally be useful in clustering.

A word2vec model is a neural network trained to create a dense vector with fixed dimension for each token in a corpus. While a pre-trained word2vec model is available for Twitter data (Godin, Vandersmissen, De Neve, & Van de Walle, 2015), we found that it did not perform well on the Twitter data sets used in this study. One issue was that many tokens in the data were out of the trained model's vocabulary, and also the semantic relationships between words may be very different on different data sets. Additionally, a pre-trained model on a large amount of Reddit data was not available. Furthermore, there are many hyper-parameters in these models so finding an ideal set of values for different data sets is a useful contribution. For these reasons, we trained our own word embedding and document embedding models.

The word2vec models used in **FR2** and **FR3** were trained with the continuous bag of words (CBOW) method (Mikolov et al., 2013), 100 dimensions, a context window of size 5 and minimum word count of 1. We tested variations of these hyper-parameters, including context window sizes ranging from 3 to 15, higher dimensions and minimum word counts. We found that the variation in performance using the three clustering evaluation measures was minimal and the chosen hyper-parameters were optimal. Some of these results make sense given the short document length of social media text. We concluded that 100 dimensions for word2vec was sufficient to represent words for short documents. The mean number of tokens per tweet was 9, and the 75th percentile was 11, so a context window of size 5 captured all the tokens of most tweets. However, we did find significant variation in the number of training epochs used for the three data sets. We report on this analysis in Section 4.1. For all other hyper-parameters, we have used default

values provided by the *gensim 3.4.0* python package (Řehůřek & Sojka, 2010).

FR2 was constructed by taking the element-wise mean of the word vectors for each token in each document, returning a dense feature vector of 100 dimensions. **FR3** was constructed by taking the *tf-idf* weighted mean of the word vectors for each word of a document. The *tf-idf* matrix used was the top 1000 term matrix by frequency constructed in **FR1**. This process excluded any word vectors that were not in the top 1000 *tf-idf* terms, although again this was tried with larger numbers of top terms for which the evaluation measures used were found to decrease. We discuss the evaluation measures used in Section 3.5.

A doc2vec model is a neural network trained to create a dense vector with fixed dimension for each document in a corpus. The doc2vec models in **FR4** were trained with 100 dimensions using the distributed bag of words method (*dbow*), a context window of size 5 and a minimum word count of 1. The distributed bag of words method was used since it can train both word vectors and document vectors in the same embedding space (Le & Mikolov, 2014), which was useful for interpreting the document embedding. As with the word2vec model, we tested variations of the hyper-parameters and found that the evaluation measures varied significantly for the number of training epochs, and different data sets had different optimal epochs. This is similar to the results of Lau and Baldwin (2016) where a *dbow* doc2vec model trained on 4.3 million words had an optimal number of epochs of 20, while the optimal number was 400 for a data set of size 0.5 million words. Lau and Baldwin (2016) also found that the optimal number of dimensions was 300 and window size was 15. The lower optimal values for our method are likely due to the short document lengths of OSN data, as well as the lower word count of our data sets, especially the Twitter data.

3.4. Clustering methods

For the clustering methods, we have selected four techniques commonly used in the literature (Klinczak & Kaestner, 2016; Naik et al., 2015) which also gave comparable results on our data sets. Firstly, we applied a k-means clustering algorithm (**CM1**) using the Euclidean metric and a maximum of 100 iterations. The algorithm was run multiple times over the data with varying random seeds. **CM2** refers to the k-medoids algorithm. For this we used the *pyclustering 0.8.2* python package with starting centroids sampled according to a uniform distribution. Both k-means and k-medoids clustering were used in Klinczak and Kaestner (2016). For **CM3** we applied an hierarchical agglomerative clustering algorithm with the Euclidean metric and Ward linkage. Hierarchical agglomerative clustering was used in Ferrara et al. (2013) to cluster a similarity matrix. For **CM4** we used a Non-negative Matrix Factorisation (NMF) algorithm, for which we used the default parameters in the *sklearn 0.19.1* package. NMF has seen multiple applications for topic modelling in OSN data (Godfrey et al., 2014; Klein et al., 2018). For the clustering methods and the LDA model, we set the number of clusters or components to be equal to the number of unique labels in the evaluation data. In line with Klinczak and Kaestner (2016), we tested the DBSCAN clustering algorithm with a range of hyper-parameters but found that it delivered poor performance for all feature representations. The documents would either be grouped into an outlier cluster, or a large number of very small clusters. A possible reason for this is that the feature representations are high dimensional and sparse, so may not cluster well using density based approaches.

The **LDA** topic model was trained with 10 passes, chunk size of 10,000 and updated every record. We again used the default values for other hyper-parameters in the *gensim 3.4.0* package. We included this method since it is commonly used in document clustering and topic modelling. To assign a topic label to each document, we chose the topic with the highest probability.

3.5. Evaluation measures

Measures used for evaluating document clustering methods typically fall into two categories, intrinsic and extrinsic measures. Intrinsic measures, such as measures of cluster separation and cohesion, do not require a ground truth label. Such measures describe the variation within clusters and between clusters. However, they are dependent on the feature representations used, so do not give comparable results for methods which use different feature sets. Extrinsic measures require a ground truth label, but can be compared across methods. Common extrinsic measures include precision, recall and F1 (Naik et al., 2015), but these are dependent on the ordering of cluster labels to ground truth labels which is a problem with a large number of labels. Measures such as the mutual information and Rand index are more appropriate in this case as they are independent of the absolute values of the labels.

Mutual information is a measure of the mutual dependence between two discrete random variables. It quantifies the reduction in uncertainty about one discrete random variable given knowledge of another. High mutual information indicates a large reduction in uncertainty. For two discrete random variables X and Y with joint probability distribution $p(x, y)$, the mutual information, $MI(X, Y)$, is given by

$$MI(X, Y) = \sum_{y \in Y} \sum_{x \in X} \log\left(\frac{p(x, y)}{p(x)p(y)}\right).$$

A commonly used measure is the normalised mutual information (NMI), which normalises the MI to take values between 0 and 1 with 0 representing no mutual information and 1 being agreement. This is useful to compare results across methods and studies. NMI is given as follows.

$$NMI(X, Y) = \frac{MI(X, Y)}{\sqrt{H(X)H(Y)}},$$

where $H(X)$ and $H(Y)$ denote the marginal entropies, given by

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log(p(x_i)).$$

The Rand index is a pair counting measure for similarity between the labels and clusters. It also takes values between 0 and 1, with 0 representing a random labelling and 1 representing identical labels. Given a set of elements $S = \{o_1, \ldots, o_n\}$ and two partitions of S to compare, $X = \{X_1, \ldots, X_r\}$ and $Y = \{Y_1, \ldots, Y_s\}$, the Rand index represents the frequency of times the partitions X and Y are in agreement over the total number of observation pairs. Mathematically the Rand index, *RI*, is given by

$$RI(X, Y) = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}},$$

where a represents the number of pairs of elements in S that are in the same subset in X and the same subset in Y, and b represents the number of pairs of elements in S that are in different subsets of X and different subsets of Y. Values a and b together give the number of times the partitions are in agreement. The value c represents the number of pairs of elements in S that are in the same subset of X and different subsets of Y, and d gives the number of pairs of elements in S that are in different subsets of X and the same subset of Y.

For extrinsic clustering evaluation measures to be useful for comparison across methods and studies, such measures need a fixed bound and a constant baseline value. Both the NMI and the RI are scaled to have values between 0 and 1, so satisfy the first condition. However, it has been shown that both measures increase monotonically with the number of labels, even with an arbitrary cluster assignment (Vinh, Epps, & Bailey, 2010). This is because both the mutual information and Rand index do not have a constant baseline, implying that these measures are not comparable across clustering methods with different numbers of clusters. To account for this, adjusted versions of the MI and RI have been proposed. The adjusted rand index, ARI, adjusts the RI by its expected value:

$$ARI(X, Y) = \frac{RI(X, Y) - E\{RI(X, Y)\}}{\max\{RI(X, Y)\} - E\{RI(X, Y)\}}$$

where $E\{RI(X, Y)\}$ denotes the expected value of $RI(X, Y)$. The ARI takes values between 0 and 1, with 1 representing identical partitions, and is adjusted for the number of partitions in X and Y. In a similar way, the adjusted mutual information, AMI, is given by

$$AMI(X, Y) = \frac{MI(X, Y) - E\{MI(X, Y)\}}{\max\{H(X), H(Y)\} - E\{MI(X, Y)\}},$$

where $E(MI(X, Y))$ represents the expected value of the MI (Vinh et al., 2010). The AMI takes values between 0 and 1, with 1 representing identical partitions, and is adjusted for the number of partitions used. The best measures to ensure a comparable evaluation are then the AMI and the ARI. The next question is around how these two measures compare to each other. By developing theory regarding generalised information theoretic measures, Romano, Vinh, Bailey, and Verspoor (2016) concluded that the AMI is the preferable measure when the labels are unbalanced and there are small clusters, while the ARI should be used when the labels have large and similarly sized volumes.

In this paper, we report the AMI, ARI and the NMI measures. Many previous studies have reported the NMI measure, so for comparison purposes we include it in our evaluation. Given the data and methods of this study, it is likely that the ARI is more appropriate then the AMI as Tables 2 and 4 show that the distribution of documents across labels is relatively balanced. We still include the AMI since it is interesting to see how much the results may differ from the NMI.

Due to the short and noisy nature of the data sets used in this study, we examined the effect of different random seeds on performance. We ran each method 20 times with different random seeds, calculated the mean of the NMI, AMI and ARI, and plotted the distributions of these measures.

4. Results

In this section we present the results of our analysis. We first describe the results on the optimal number of epochs for the word2vec and doc2vec embedding representations, applied to all three data sets. We then evaluate the performance of all the methods. Lastly, we discuss methods for the interpretation of the topics using the doc2vec feature representation.

4.1. Optimal training epochs for embedding models

A key hyper-parameter for training neural network models is the number of epochs. Too many epochs and the model may overfit to the data, too few and performance may be poor. We first explored the performance change of the mean word2vec models (**FR2** and **FR3**) and the doc2vec model (**FR4**) with the number of epochs. These results provide guidance for studies where a ground truth topic label is not present. We used k-means clustering (**CM1**) for the clustering method as it gave the best results for the embedding representations. For each epoch value between 25 and 300, with increments of 25, we trained the models 20 times using different random seeds and evaluated against the ground truth labels. This was done for all three data sets. Table 6 summarises the optimal epoch results by method and data set. The plots for this analysis on the #Auspol Twitter data are shown in Fig. 2(a) and on the RepLab 2013 data in Fig. 2(b). The results for the Reddit data are shown in Fig. 3. To save space we only evaluated the AMI and the ARI measures on the Reddit data. This is because the AMI typically gives similar results as NMI, but is chance adjusted.

For the #Auspol data in Fig. 2(a), it is clear that doc2vec gave the best results and had a peak in performance at around 75 epochs.

Table 6
Optimal number of training epochs for word2vec and doc2vec methods on the three data sets.

Data set	doc2vec	wtd. word2vec	unwtd. word2vec
Twitter #Auspol	75	250	250
Twitter RepLab 2013	300	200	200
Reddit: 1–101	175	75	50
Reddit: 101–200	150	100	200
Reddit: 201–500	100	50	50
Reddit: 501 +	50	25	25

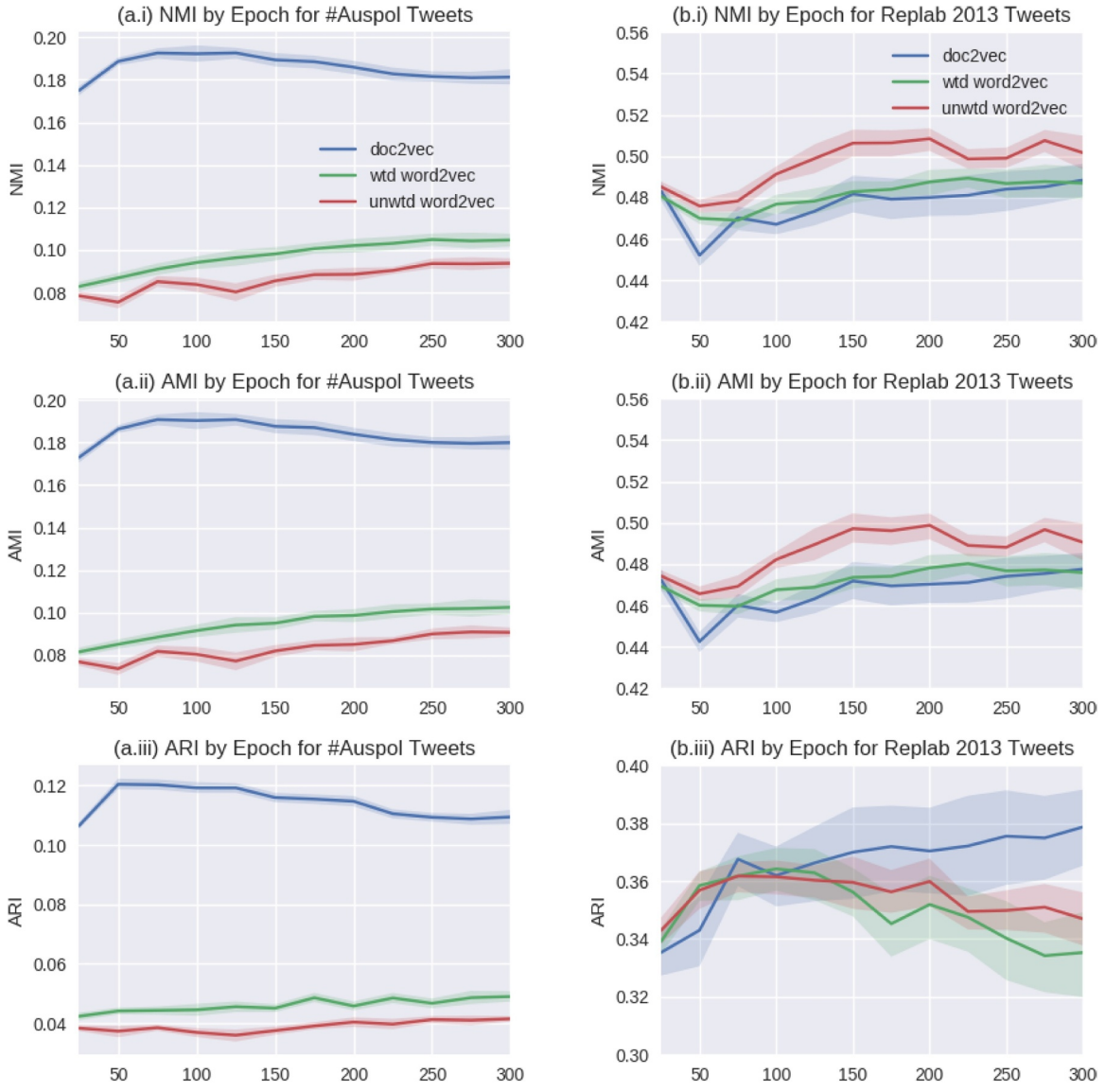
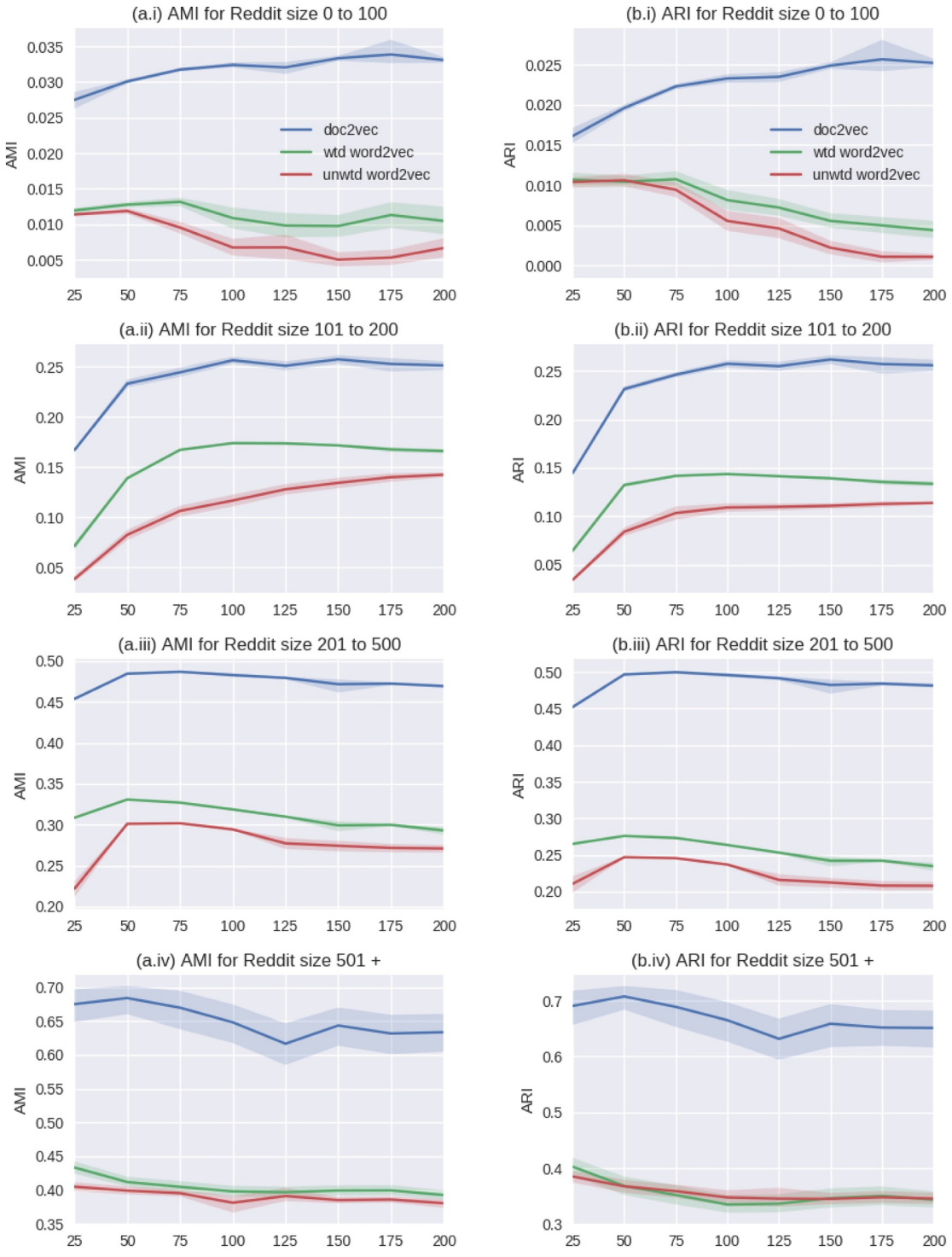


Fig. 2. Plot of the three evaluation measures (vertical axes) by training epoch (horizontal axes) for 20 runs of the word2vec and doc2vec representations on Twitter data using k-means clustering. (a) shows the results on the #Auspol Twitter data and (b) shows the results on the RepLab 2013 Twitter data. 95% confidence bands based on varying random seeds are shown.



(a) AMI measure (b) ARI measure

Fig. 3. Plots of the AMI and ARI evaluation measures (vertical axes) by training epoch (horizontal axes) for 20 runs of the word2vec and doc2vec representations on the Reddit data sets using k-means clustering. Different Reddit data sets by size range are given along the rows. Column (a) shows the AMI results and (b) shows the ARI results. 95% confidence bands based on varying random seeds are shown.

The word2vec methods generally delivered better performance with more epochs, with a maximum value around 250. The *tf-idf* weighted mean word2vec method performed better than the unweighted mean word2vec method, and its performance increased more smoothly than the unweighted method. There was also not much variation over seeds as the 95% confidence bands are narrow.

On the RepLab 2013 data in Fig. 2(b) the results were quite different. The unweighted mean word2vec method gave the best performance on the NMI and AMI measures. However, on the ARI measure both word2vec methods suffered drops in performance after 100 epochs while the doc2vec method improved. This could be caused by some over-fitting of the word2vec models on the data, which is likely since the RepLab 2013 data was much smaller than the #Auspol data. The ARI measure is also the preferred measure where the labels have large volumes and are balanced (Romano et al., 2016). This data set was relatively balanced (given in Table 3), so the ARI is the more appropriate performance measurement than the NMI and AMI. Overall on the RepLab 2013 data, the optimal number of epochs for the word2vec methods was 200, while the doc2vec method had an optimal value of 300. The higher number of optimal epochs for the doc2vec method is not surprising given that it is also training document vectors, so has more parameters than word2vec.

Turning to the results on the four Reddit data sets in Fig. 3, the doc2vec method again gave the best performance. In addition, there is an evident pattern with doc2vec where shorter documents required more training epochs to reach optimal performance. For documents with less than 100 characters, the performance of doc2vec with k-means clustering improved up to around 250 epochs. This dropped to 150 epochs for documents with 101–200 characters, then 150, 100 and 50 for the larger document length data sets in increasing order. This observed pattern aligns to the results of Lau and Baldwin (2016), confirming that doc2vec models require less training epochs on larger documents.

For the word2vec methods, the *tf-idf* weighted mean word vector method gave better performance than the unweighted mean method. This aligns with results in previous studies (Billah Nagoudi et al., 2017). On the shortest document range, both methods showed little performance improvement with more training, but then a drop in both measures at 75 epochs for the weighted word2vec method and 50 for the unweighted method. One possible explanation for this drop is that averaging word vectors may only make sense above a threshold of words. For this size range, the average number of words per document is 4.5, which might be too low. On the 101 to 200 character length documents, the weighted word2vec method gave better performance but also required fewer training epochs. These results also look similar to the results on the Twitter data sets, which typically have a similar character length range. On the largest documents, both methods required 25 or less epochs to reach optimal performance.

Through this analysis, it is clear that the doc2vec method consistently gave improved performance over averaged word2vec methods, except in the case where the data set had a low number of documents. Furthermore, the number of training epochs for doc2vec in general was inversely proportional to the document size, with more epochs required to reach optimal performance on smaller document sizes. Doc2vec also required more training epochs than word2vec, in general. However, these relations were not observed for the #Auspol Twitter data where the doc2vec optimal epoch number was 75, below the word2vec optimal epochs of 200. The optimal number of doc2vec epochs on the RepLab 2013 data was much higher at 300. An explanation might be that while the doc2vec model improved on its internal loss function with more training epochs on the #Auspol data, these improvements did not lead to better performance on the clustering task. This is likely because of the hashtag labels used, which may have some overlapping contributing terms. For the word2vec methods, in general weighting by *tf-idf* scores gave a performance lift and required fewer training epochs. However, care should be taken with the number of epochs given the low peak on the shortest Reddit documents.

4.2. Performance evaluation with clustering measures

In this section we provide the mean evaluation measures for the four feature representations with the four clustering methods, and

Table 7
Performance evaluation of the feature representation and clustering methods on #Auspol Twitter data with the Normalised Mutual Information (NMI), Adjusted Mutual Information (AMI), and Adjusted Rand Index (ARI) measures.

Feature representation	Clustering	NMI	AMI	ARI
doc2vec	Hierarchical	0.165	0.154	0.059
	k-means	**0.193**	**0.191**	**0.120**
	k-medoids	0.107	0.105	0.064
	NMF	0.102	0.100	0.056
wtd word2vec	Hierarchical	0.088	0.079	0.021
	k-means	0.105	0.102	0.047
	k-medoids	0.043	0.016	0.001
	NMF	0.062	0.058	0.030
unwtd word2vec	Hierarchical	0.085	0.076	0.020
	k-means	0.094	0.090	0.041
	k-medoids	0.043	0.019	0.001
	NMF	0.058	0.054	0.025
TF-IDF	Hierarchical	0.163	0.085	0.013
	k-means	0.114	0.070	0.014
	k-medoids	0.079	0.028	0.004
	NMF	0.132	0.110	0.032
LDA	LDA	0.043	0.041	0.021

Table 8
Performance evaluation of the feature representation and clustering methods on RepLab 2013 Twitter data with the NMI, AMI and ARI measures.

Feature representation	Clustering	NMI	AMI	ARI
doc2vec	Hierarchical	0.449	0.437	0.313
	k-means	0.488	0.478	**0.379**
	k-medoids	0.290	0.278	0.215
	NMF	0.261	0.249	0.152
wtd word2vec	Hierarchical	0.506	0.491	0.330
	k-means	0.488	0.478	0.352
	k-medoids	0.421	0.404	0.274
	NMF	0.401	0.384	0.266
unwtd word2vec	Hierarchical	**0.519**	**0.507**	0.347
	k-means	0.508	0.499	0.360
	k-medoids	0.435	0.414	0.278
	NMF	0.425	0.407	0.286
TF-IDF	Hierarchical	0.466	0.417	0.203
	k-means	0.450	0.379	0.179
	k-medoids	0.192	0.075	0.011
	NMF	0.437	0.427	0.348
LDA	LDA	0.180	0.169	0.140

the LDA model, for each method with 20 different seeds on each data set. We also include distribution plots to illustrate the variability in performance.

Table 7 provides the mean for each of the three evaluation measures for each method on the #Auspol Twitter data set. We set the optimal number of epochs to be 75 for the doc2vec methods and 250 for the word2vec methods. It is clear from this table that the doc2vec feature representation with k-means clustering outperformed the other methods on all three evaluation measures, particularly on the ARI. Hierarchical clustering gave close scores for NMI and AMI, but much lower ARI. For both doc2vec and word2vec feature representations, NMF performed poorly. The performance of k-medoids clustering was similar to NMF. For the word2vec representations, k-means clustering also gave the best performance.

An interesting observation is that some methods had a relatively large drop in score between the NMI and AMI measures, indicating that the chance adjustment of the AMI is important. The *tf-idf* representation is the most effected by this. For instance, the *tf-idf* matrix with hierarchical clustering gave a high NMI of 0.163, well ahead of the word2vec methods, but an AMI of 0.085. Comparatively, doc2vec and the word2vec methods had smaller drops. As discussed earlier, the AMI and ARI are more appropriate evaluation measures than NMI due to their adjustment for chance. On this data set, the ARI is more appropriate as the volume of tweets per hashtag label are relatively similar. The doc2vec representation with k-means clustering therefore far outperformed the other methods.

Table 8 shows the mean results for the RepLab 2013 Twitter data set with the doc2vec model trained with 300 epochs and the word2vec methods trained with 200 epochs. Overall the performance is much higher than in the #Auspol data, which is explained by the RepLab 2013 data having expertly annotated topics which are more distinct. On the ARI score, the doc2vec method with k-means clustering performed best but the unweighted word2vec method with hierarchical clustering gave higher performance for the NMI and AMI measures. One explanation for this is that the small size of this data set is insufficient for the embedding representations to accurately be trained, so further training does not necessarily lead to higher clustering performance. This is reflected in the sharp drops evident in Fig. 2(b.i) and (b.ii).

To examine the variability from the mean measurements, we plot the distributions for the feature representation methods with the best performing clustering algorithm and the LDA topic model. Fig. 4 shows the distributions for the three evaluation measures over the #Auspol (a) and RepLab 2013 (b) Twitter data sets. In Fig. 4(a), the doc2vec method with k-means clustering was distinctly ahead of the other methods on all three measures. There was also significant overlap between the results for the two word2vec methods, indicating that multiple runs are required when the scores are close. Note that the *tf-idf* method with hierarchical clustering did not show on the plot since both algorithms are deterministic, so every run had the same result.

For the RepLab 2013 data set in Fig. 4(b), the word2vec methods again showed significant overlap, with the doc2vec method performing in a lower range. It is interesting to note that the doc2vec method showed two close peaks. These peaks are most significant for the NMI and AMI measures, but also present for ARI. This likely indicates that the doc2vec method optimised to local minima during training, resulting in poor performance for some of the runs over random seeds. Given that there was a large gap between the higher performance of doc2vec compared to the word2vec methods on the #Auspol data and close performance between word2vec and doc2vec on RepLab 2013, the word2vec methods handled the smaller RepLab 2013 data set better than doc2vec. This may be because there weren't enough data points in the RepLab 2013 data set to optimally train the doc2vec representation. Nevertheless, doc2vec still gave the best performance on the ARI measure for both Twitter data sets.

Lastly, we provide results from running the methods over the Reddit data. Fig. 5 shows the NMI (a), AMI (b) and ARI (c) values for the methods on the Reddit data sets. The horizontal axis compares the results for the document length data partitions. Only the best performing clustering method is displayed for each feature representation. The mean scores of the evaluation measures for each of the methods are given in Table 9 for document length ranges 1–100 and 101–200, and in Table 10 for document length ranges 201–500 and 501 or greater. It is clear from these plots and mean results that the doc2vec method delivered the best performance on all four

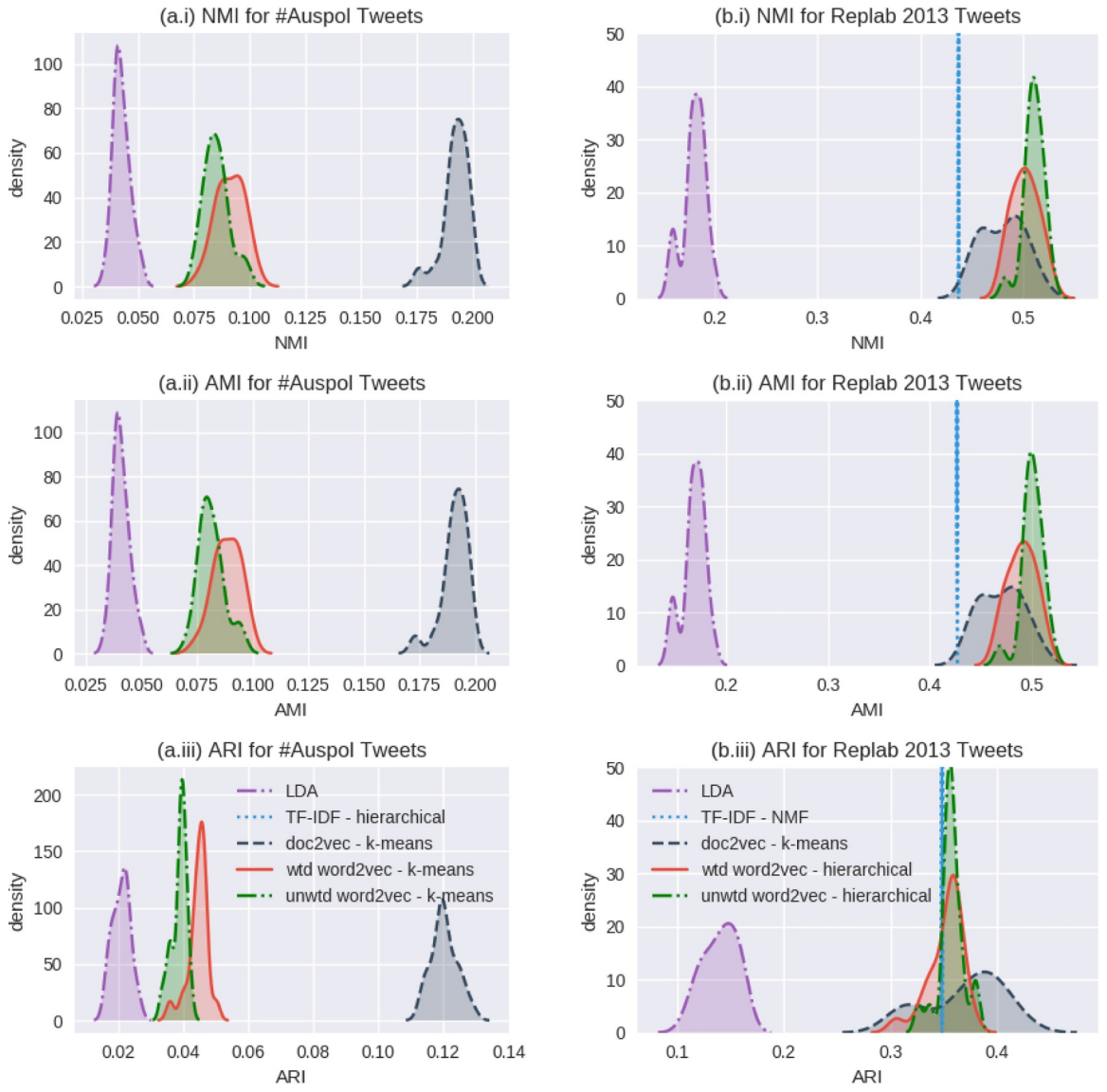


Fig. 4. Density plots of the three evaluation measures (horizontal axes) over random seeds for the four feature representations with the best performing clustering algorithm, with LDA for comparison. (a) shows the results on the #Auspol Twitter data and (b) shows the results on the RepLab 2013 Twitter data.

data sets by size range. This finding corroborates the results from the #Auspol Twitter data set. The *tf-idf* weighted mean word2vec method consistently delivered a performance lift compared to the unweighted mean word2vec method. Interestingly, the *tf-idf* methods and the LDA model only gave comparable performance to the word2vec methods on the last size range, with number of characters greater than 500.

4.3. Topic interpretation

It is clear that the doc2vec model with k-means clustering delivered the best performance on the #Auspol Twitter data set and the Reddit data sets, as well as the RepLab 2013 Twitter data set based on the ARI measure only. However, the usefulness of a topic discovery model depends on how interpretable the resulting topics are. In this section we aim to address this question through a deeper analysis of the resulting clusters from the doc2vec representation with k-means clustering.

We consider firstly the results on the #Auspol data, where we analysed the extent to which the document clusters aligned to the

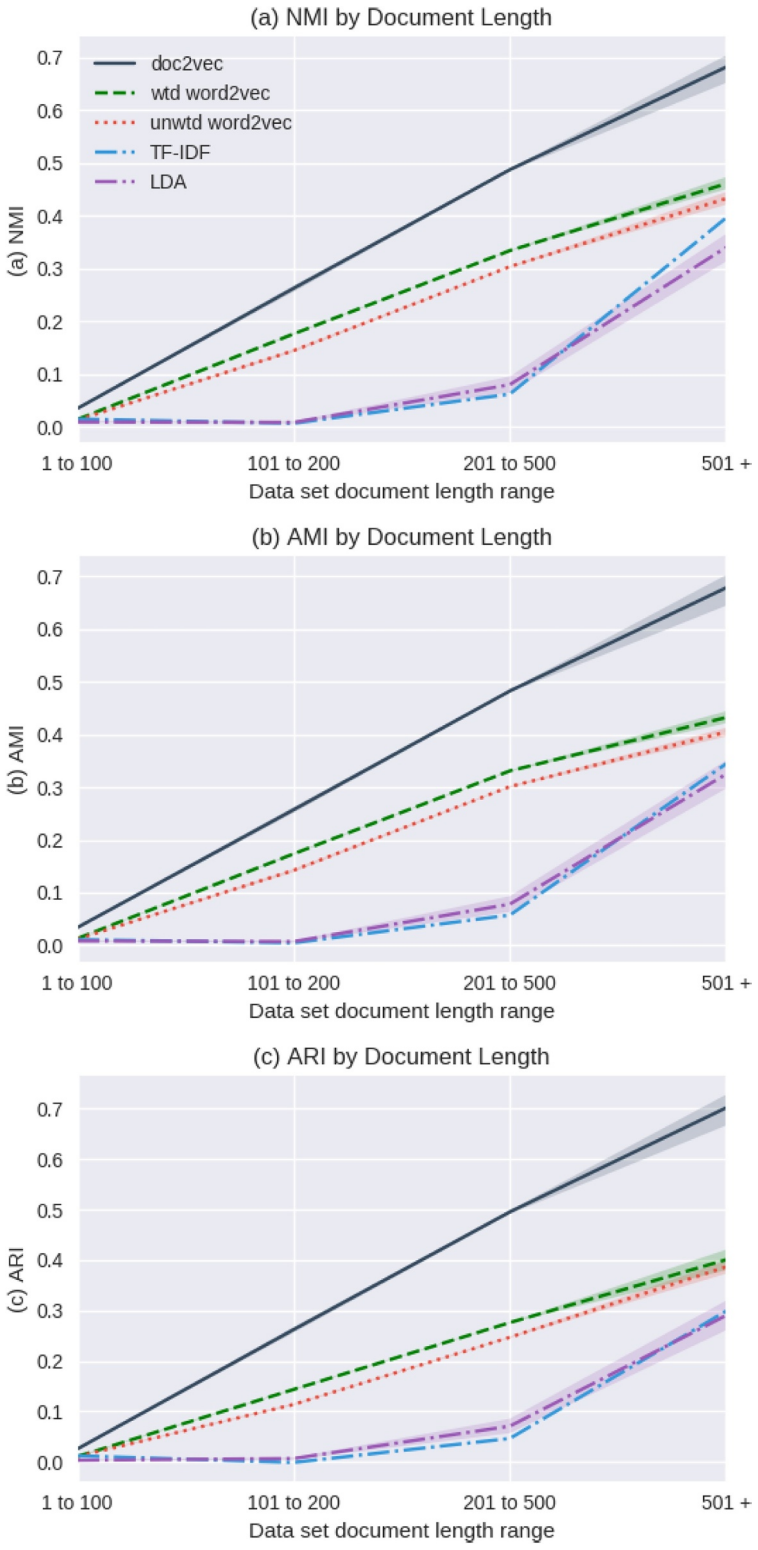


Fig. 5. Plot of the three evaluation measures over random seeds for the methods with the best performing clustering method on Reddit data with varying document lengths in characters. (a) plots the NMI, (b) plots the AMI and (c) plots the ARI.

Table 9
Performance evaluation on the Reddit data for each method by for document length ranges 1–100 and 101–200 characters.

Document character length	Feature representation	Clustering	NMI	AMI	ARI
1–100	**doc2vec**	Hierarchical	0.029	0.027	0.017
		k-means	**0.034**	**0.034**	**0.026**
		k-medoids	0.012	0.011	0.004
		NMF	0.030	0.023	0.015
	wtd word2vec	Hierarchical	0.013	0.012	0.010
		k-means	0.014	0.013	0.011
		k-medoids	0.007	0.003	0.000
		NMF	0.010	0.009	0.000
	unwtd word2vec	Hierarchical	0.011	0.011	0.011
		k-means	0.012	0.012	0.011
		k-medoids	0.007	0.006	0.000
		NMF	0.012	0.011	0.010
	TF-IDF	Hierarchical	0.009	0.003	0.000
		k-means	0.005	0.002	0.000
		k-medoids	0.005	0.001	0.000
		NMF	0.014	0.011	0.012
	LDA	LDA	0.009	0.009	0.003
101–200	**doc2vec**	Hierarchical	0.115	0.111	0.067
		k-means	**0.262**	**0.257**	**0.262**
		k-medoids	0.018	0.006	0.001
		NMF	0.127	0.096	0.027
	wtd word2vec	Hierarchical	0.112	0.101	0.032
		k-means	0.176	0.174	0.144
		k-medoids	0.036	0.016	0.001
		NMF	0.116	0.100	0.033
	unwtd word2vec	Hierarchical	0.086	0.079	0.027
		k-means	0.144	0.142	0.114
		k-medoids	0.020	0.013	0.008
		NMF	0.089	0.071	0.015
	TF-IDF	Hierarchical	0.009	0.003	0.000
		k-means	0.005	0.004	0.002
		k-medoids	0.008	0.000	0.000
		NMF	0.006	0.005	0.000
	LDA	LDA	0.008	0.007	0.007

label hashtags. On the #Auspol data, our ground truth topic labels were the top 13 distinct hashtags, which were removed from the text prior to feature generation and clustering. These hashtags can therefore be considered as latent tokens. We first identified the top three topic labels (hashtags) by frequency for each cluster. For comparison, we created a *tf-idf* matrix from the original data using all the hashtags, including the topic hashtags, and excluded all other tokens. We then extracted the three hashtags with the highest *tf-idf* scores for each cluster and compared to the top three topic label hashtags. Table 11 outlines the results. The top topic matches to the top hashtag for every cluster. Out of 39 top topics for the 13 clusters, only 7 contained different hashtags (highlighted in bold text). There were also two clusters where the order was adjusted. We conclude that the doc2vec clustering has accurately captured the structure of the latent label hashtags.

Another way of looking at the quality of the clustering is to analyse the overlap between ground truth labels and clusters. In the interest of space, we considered the Reddit data sets which contained only 5 topics and chose the data set with document size between 101 and 200 characters for consistency with the Twitter data sets. We then analysed the confusion matrix for the doc2vec features with k-means clustering against the ground truth labels, the subreddit pages. The results are shown in Table 12. It is apparent that the first cluster grouped most of the parent comments from the subreddit page 'NFL' and the second cluster grouped strongly around 'pcmasterrace'. These pages clearly represent distinct topics. Clusters 3 and 4 grouped well around 'news' and 'movies' respectively, but cluster 5 is divided primarily between 'relationships' and 'news'.

To further interpret the topics on this Reddit data set, we analysed the top words by cluster. For each cluster we calculated the centroid as the mean of the doc2vec representations of each document in the cluster. Since the trained doc2vec model produced document embeddings in the same space as word embeddings, we calculated the cosine similarity between the cluster centroids and the words. The idea behind this was that words closer to the cluster centroid may be representative of the cluster. However, this approach doesn't account for the frequency of words appearing in each cluster, or the relative frequency of the words across the clusters. To incorporate this information, we pooled all the documents in each cluster and calculated a *tf-idf* matrix. We then created a combined score for each word and cluster from the sum of the cosine similarity and the *tf-idf* score. Table 13 shows the top 10 words per cluster ordered by this method. It is clear that this method extracts very specific terms related to the main subreddit pages, particularly for clusters 1 and 2.

Table 10
Performance evaluation on the Reddit data for each method by document length ranges 201–500 and 501 or greater.

Document Character Length	Feature Representation	Clustering	NMI	AMI	ARI
201–500	**doc2vec**	Hierarchical	0.261	0.254	0.212
		k-means	**0.487**	**0.483**	**0.496**
		k-medoids	0.037	0.010	0.002
		NMF	0.194	0.128	0.044
	wtd word2vec	Hierarchical	0.265	0.246	0.142
		k-means	0.333	0.331	0.276
		k-medoids	0.174	0.172	0.150
		NMF	0.247	0.226	0.133
	unwtd word2vec	Hierarchical	0.227	0.200	0.084
		k-means	0.303	0.301	0.247
		k-medoids	0.106	0.103	0.081
		NMF	0.208	0.183	0.092
	TF-IDF	Hierarchical	0.103	0.061	0.015
		k-means	0.095	0.085	0.044
		k-medoids	0.014	0.013	0.007
		NMF	0.062	0.057	0.046
	LDA	LDA	0.080	0.079	0.071
501 +	**doc2vec**	Hierarchical	0.532	0.518	0.499
		k-means	**0.686**	**0.684**	**0.708**
		k-medoids	0.094	0.037	0.007
		NMF	0.331	0.255	0.154
	wtd word2vec	Hierarchical	0.465	0.400	0.327
		k-means	0.461	0.433	0.403
		k-medoids	0.353	0.330	0.283
		NMF	0.366	0.325	0.229
	unwtd word2vec	Hierarchical	0.416	0.367	0.306
		k-means	0.433	0.405	0.385
		k-medoids	0.336	0.322	0.290
		NMF	0.290	0.242	0.159
	TF-IDF	Hierarchical	0.304	0.244	0.199
		kmeans	0.431	0.382	0.323
		kmedoids	0.042	0.007	0.001
		NMF	0.396	0.344	0.299
	LDA	LDA	0.341	0.326	0.291

Table 11
Top three topic labels and top three hashtags for each cluster. Note that the topic labels did not appear in the clustering data, but were mostly recovered in order when we created a *tf-idf* matrix for tweets pooled by cluster and selected the three hashtags with the highest scores. Differences between the top three topic labels and top three hashtags are highlighted in bold.

Cluster	**Top three topic labels**	**Top three *tf-idf*score hashtags**
1	#nbn, #lnp, #insiders	#nbn, #lnp, #insiders
2	#uspoli, #insiders, **#turnbull**	#uspoli, #insiders, **#trump**
3	#insiders, #lnp, #qldpol	#insiders, #lnp, #qldpol
4	#qldpol, #insiders, #lnp	#qldpol, #insiders, #lnp
5	#politas, **#qldpol, #lnp**	#politas, **#utas, #discover**
6	#qldpol, #qanda, **#trump**	#qldpol, **#politas**, #qanda
7	#insiders, #lnp, #qldpol	#insiders, #lnp, #qldpol
8	#qldpol, #stopadani, #springst	#qldpol, #stopadani, #springst
9	#lnp, #trump, **#uspoli**	#lnp, #trump, **#insiders**
10	#qanda, #insiders, **#qldpol**	#qanda, #insiders, **#sayitwithstickers**
11	#marriageequality, #politas, **#lnp**	#marriageequality, **#equalitycampaign**, #politas
12	#qldpol, #stopadani, #qanda	#qldpol, #qanda, #stopadani
13	#climatechange, #qldpol, #stopadani	#climatechange, #qldpol, #stopadani

Table 12
Confusion matrix for the doc2vec representation with k-means clustering method on Reddit data with size range between 101 and 200 characters.

Subreddit page	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
NFL	**1351**	60	298	273	395
pcmasterrace	78	**1295**	215	185	260
News	93	89	**952**	204	538
Movies	89	50	152	**767**	226
Relationships	32	37	116	48	**557**

Table 13
Top 10 words per cluster based on combined embedding similarity score in embedding space and *tf-idf* score.

Cluster	**Top topic**	**Top 10 words**
1	NFL	talent, flacco, quarterback, tds, sb, wrs, roster, dolphins, tackle, foles
2	pcmasterrace	install, ps4, r9, mobo, gpus, i5, os, msi, processor, asus
3	News	federal, manslaughter, district, homicide, economic, isis, china, labor, upper, toke
4	Movies	avengers, joss, horror, arnold, cinematography, rewatch, australian, doof, boobs, mcx
5	Relationships	abusive, mentality, react, rdj, xanax, marriage, heaven, meeting, section, subjective

5. Discussion

Throughout this study it has become clear that for clustering OSN text data into topics, in general doc2vec feature representations combined with k-means clustering gave the best performance compared to any of the other methods. However, the cases where this method did not perform as well require discussion. On the RepLab 2013 Twitter data set, the doc2vec method gave performance below that of the mean word2vec methods for the NMI and AMI measures, but gave the best performance for ARI after 100 epochs of training. Further to this, the unweighted mean word2vec method performed better than the *tf-idf* weighted mean word2vec method on this data. Both of these results are different from the results on the other two data sets. The results on the #Auspol and the Reddit data with document length between 101 and 200 characters indicate that it is not the size of each document that is the issue on the RepLab data, but it is most likely that the volume of data used was not sufficient to accurately train the doc2vec model. The implication is that doc2vec models should be trained on data volumes greater than three thousand or so. Interestingly, the Reddit data with length between 101 and 200 characters only consisted of 8360 documents and doc2vec performed very well, although Reddit comments may be quite different to tweets in the terms used.

Another interesting observation is that on the #Auspol Twitter data, the *tf-idf* matrix with NMF gave better performance on the NMI and AMI measures than the best clustering for both word2vec methods, although a lower score for the ARI. On the RepLab 2013 data, the word2vec methods performed better on NMI and AMI, but the *tf-idf* method was very close on ARI. However, on the Reddit data the *tf-idf* method gave very low performance until the document size was greater than 200 characters. This indicates that topics from Twitter text may rely heavily on keywords since the *tf-idf* clustering performs comparatively well, which is not surprising given the use of user mentions and hashtags. The doc2vec method represented this information more effectively on the #Auspol data than the other feature methods. Assigning a heavier weighting for hashtags and user mentions for the doc2vec model might give improved performance on Twitter data.

Two useful results stand out from this study based on the Reddit data. The first was that the optimal number of training epochs for doc2vec is inversely proportional to the average length of the documents. This result provides some guidance for future studies using OSN data. Unfortunately this result was not consistent with the results on the #Auspol data, which may be due to the topic labels themselves not being clearly distinct. There is an ongoing challenge with using Twitter data as manually labelling topics is time consuming and prone to error, and the number of retrievable tweets diminishes over time. The result is consistent with the RepLab 2013 Twitter data, but as discussed already the data volume was small. The second result is that the performance of the doc2vec method increased with the length of the documents. The method gave high performance for the longest Reddit comments, so should give good results applied to text data from OSN platforms in general.

Improving embedding representations of OSN documents can be useful for several natural language processing tasks. Such representations at the document level can provide high quality feature matrices to be used by other machine learning systems. An example application is for sentiment analysis (Lee, Jin, & Kim, 2016). In addition, it has been shown previously that pre-training the word vectors used by doc2vec can provide a performance lift in several natural language processing tasks (Lau & Baldwin, 2016). Pre-training both word vectors and document vectors for large volumes of OSN data could then provide a performance lift on applications focused on specific samples of data. For instance, pre-trained document vectors could be used in streaming document classification or clustering applications. In addition, such methods could be applied in other domains where data can be modelled as documents with a small number of tokens. For example, embedding models are seeing applications on electronic health record data (Choi et al., 2016). In this instance, medical codes are treated as tokens and embedding models can then be used to capture information about relationships between diseases and treatments, and be used in subsequent prediction or clustering tasks.

6. Conclusion and future work

In this study we showed the different performance of several document clustering and topic modelling methods on social media text data. Our results have demonstrated that document and word embedding representations of online social network data may be used effectively as a basis for document clustering. These methods outperformed traditional *tf-idf* based approaches and topic modelling techniques. Furthermore, doc2vec and *tf-idf* weighted mean word embedding representations delivered better results than simple averages of word embedding vectors in document clustering tasks. We also demonstrated that k-means clustering provided the best performance with doc2vec embeddings.

Through applying these methods over the Reddit data set split by document length ranges, we outlined two key results for clustering doc2vec embeddings. Firstly, the optimal number of training epochs is in general inversely proportional to the character length range of the documents. Secondly, doc2vec embeddings with k-means clustering provide good performance over all the

document length ranges in the Reddit data used. These results indicate that this method should perform well on most OSN text data.

To interpret the resulting clusters from these methods, we developed a top term analysis based on combining *tf-idf* scores and word vector similarities. We demonstrated that this method can provide a representative set of keywords for a topic cluster. We also showed that the doc2vec embedding with k-means clustering may successfully recover latent hashtag structure in Twitter data.

We plan several extensions to this work. Firstly, the doc2vec embeddings combined with k-means clustering can be applied readily to any social media text data. In further applications we intend to demonstrate the usefulness of this method in defining and interpreting dynamic topics in a streaming fashion. Secondly, this method may be extended to incorporate additional data available in social networks, and specifically from Twitter user and network data. Thirdly, recent developments in the applications of neural embedding and deep learning techniques, such as contextualised embedding models (Peters et al., 2018), Latent LSTM Allocation (Zaheer, Ahmed, & Smola, 2017) and deep learning based clustering models (Min et al., 2018) may be applied to deliver improved feature representations or document clusterings. Word and document embeddings may also be used as pre-trained initial layers in deep clustering and topic modelling techniques.

Acknowledgements and declarations

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2019.04.002.

References

Alghamdi, R., & Alfalqi, K. (2015). A survey of topic modeling in text mining. *International Journal of Advanced Computer Science and Applications, 3*(7), 774–777.
Alnajran, N., Crockett, K., McLean, D., & Latham, A. (2017). *Cluster analysis of twitter data: A review of algorithms. Proceedsings of the 9th international conference on agents and artificial intelligence – volume 2: ICAART, INSTICC*. SciTePress239–249.
Amigó, E., Carrillo de Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Martín, T., et al. (2013). *Overview of RepLab 2013: Evaluating online reputation monitoring systems. Proceedings of thefourth international conference of the CLEF initiative*333–352.
Bakshy, E., Rosenn, I., Marlow, C., & Adamic, L. (2012). *The role of social networks in information diffusion. Proceedings of the 21st international conference on world wide web series WWW '12*. New York, NY, USA: ACM519–528.
Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research, 3*, 1137–1155.
Billah Nagoudi, E. M., Ferrero, J., & Schwab, D. (2017). *LIM-LIG at SemEval-2017 Task1: Enhancing the semantic similarity for arabic sentences with vectors weighting. International workshop on semantic evaluations (SemEval-2017) Proceedings of the 11thinternational workshop on semantic evaluations (SemEval-2017)*125–129 Vancouver, Canada
Bisht, S., & Paul, A. (2013). Document clustering: A review. *International Journal of Computer Applications, 73*, 26–33.
Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.
Chinnov, A., Kerschke, P., Meske, C., Stieglitz, S., & Trautmann, H. (2015). *An overview of topic discovery in twitter communication through social media analytics. Proceedings of the Americas conference on information systems*1–10.
Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., et al. (2016). *Multi-layer representation learning for medical concepts. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. New York, NY, USA: ACM1495–1504.
Corrêa Júnior, E. A., Marinho, V. Q., & dos Santos, L. B. (2017). *NILC-USP at SemEval-2017 task 4: A multi-view ensemble for twitter sentiment analysis. Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*. Association for Computational Linguistics611–615.
Curiskis, S., Drake, B., Osborn, T., & Kennedy, P. Submitted. Topic labelled online social network data sets from twitter and reddit. Data In Brief.
Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., & Cohen, W. (2016). *Tweet2vec: Character-based distributed representations for social media. Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*. Association for Computational Linguistics269–274.
Fang, Y., Zhang, H., Ye, Y., & Li, X. (2014). Detecting hot topics from twitter: A multiview approach. *Journal of Information Science, 40*(5), 578–593.
Ferrara, E., JafariAsbagh, M., Varol, O., Qazvinian, V., Menczer, F., & Flammini, A. (2013). *Clustering memes in social media. Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining*. New York, NY, USA: ACM548–555.
Godfrey, D., Johns, C., Meyer, C. D., Race, S., & Sadek, C. (2014). A case study in text mining: Interpreting twitter data from world cup tweets. *CoRR*, 1–11 abs/1408.5427
Godin, F., Vandersmissen, B., De Neve, W., & Van de Walle, R. (2015). *Multimedia lab @ ACL WNUT NER shared task: Named entity recognition for twitter microposts using distributed word representations. Proceedings of the workshop on noisy user-generated text*. Association for Computational Linguistics146–153.
Guille, A., Hacid, H., Favre, C., & Zighed, D. (2013). Information diffusion in online social networks: A survey. *ACM SIGMOD Record, 42*, 17–28.
Gutman, J., & Nam, R. (2015). *Text classification of reddit postsTechnical Report*. New York University.
Ha, T., Beijnon, B., Kim, S., Lee, S., & Kim, J. H. (2017). Examining user perceptions of smartwatch through dynamic topic modeling. *Telematics and Informatics, 34*(7), 1262–1273.
Hong, L., & Davison, B. D. (2010). *Empirical study of topic modeling in twitter. Proceedings of the first workshop on social media analytics*. New York, NY, USA: ACM80–88.
Irfan, R., King, C. K., Grages, D., Ewen, S., Khan, S. U., Madani, S. A., et al. (2015). A survey on text mining in social networks. *The Knowledge Engineering Review, 30*(2), 157170.
JafariAsbagh, M., Ferrara, E., Varol, O., Menczer, F., & Flammini, A. (2014). Clustering memes in social media streams. *Social Network Analysis and Mining, 4*(1), 237.
Klein, C., Clutton, P., & Polito, V. (2018). Topic modeling reveals distinct interests within an online conspiracy forum. *Frontiers in Psychology, 9*, 1–12.
Klinczak, M., & Kaestner, C. (2016). Comparison of clustering algorithms for the identification of topics on twitter. *Latin American Journal of Computing - LAJC, 3*, 19–26.
Lau, J. H., & Baldwin, T. (2016). *An empirical evaluation of doc2vec with practical insights into document embedding generation. Proceedings of the 1st workshop on representation learning for NLP*. Association for Computational Linguistics78–86.
Le, Q. V., & Mikolov, T. (2014). *Distributed representations of sentences and documents. Proceedings of the 31th international conference on machine learning, ICML 2014, Beijing, China, 21–26 June 2014*1188–1196.
Lee, S., Jin, X., & Kim, W. (2016). *Sentiment classification for unlabeled dataset using doc2vec with jst. Proceedings of the 18th annual international conference on electronic commerce: E-commerce in smart connected world*. New York, NY, USA: ACM28:1–28:5.
Li, Q., Shah, S., Liu, X., & Nourbakhsh, A. (2017). *Data sets: Word embeddings learned from tweets and general data. Proceedings of the eleventh international conference on web and social media, ICWSM 2017, montréal, québec, canada, may 15–18, 2017*.428–436.
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781. 1301.3781

Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., & Long, J. (2018). A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access, 6*, 39501–39514.

Naik, M. P., Prajapati, H. B., & Dabhi, V. K. (2015). *A survey on semantic document clustering. 2015 IEEE international conference on electrical, computer and communication technologies (ICECCT)*1–10.

Patki, U., & Khot, D. P. (2017). A literature review on text document clustering algorithms used in text mining. *Journal of Engineering Computers and Applied Sciences, 6*(10), 16–20.

Paul, M. J., & Dredze, M. (2014). Discovering health topics in social media using topic models. *PLoS One, 9*(8), 1–11.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations. Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)*. Association for Computational Linguistics2227–2237.

Reddit (2015). r/datasets - i have every publicly available reddit comment for research. 1.7 billion comments at 250 gb compressed. any interest in this? (Accessed 19 January 2019). https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment.

Řehůřek, R., & Sojka, P. (2010). *Software framework for topic modelling with large corpora. Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*. Valletta, Malta: ELRA45–50.

Romano, S., Vinh, N. X., Bailey, J., & Verspoor, K. (2016). Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research, 17*(1), 4635–4666.

Shabunina, E., & Pasi, G. (2018). A graph-based approach to ememes identification and tracking in social media streams. *Knowledge-Based Systems, 139*, 108–118.

Steinskog, A., Therkelsen, J., & Gambäck, B. (2017). *Twitter topic modeling by tweet aggregation. Proceedings of the 21st nordic conference on computational linguistics*. Association for Computational Linguistics77–86.

Stieglitz, S., Mirbabaie, M., Ross, B., & Neuberger, C. (2018). Social media analytics challenges in topic discovery, data collection, and data preparation. *International Journal of Information Management, 39*, 156–168.

Suri, P., & Roy, N. R. (2017). *Comparison between LDA & NMF for event-detection from large text stream data. 2017 3rd international conference on computational intelligence communication technology (CICT)*1–5.

Vinh, N. X., 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.

Yang, X., Macdonald, C., & Ounis, I. (2017). Using word embeddings in twitter election classification. *Information Retrieval, 21*(2–3), 183–207.

Zaheer, M., Ahmed, A., & Smola, A. J. (2017). *Latent LSTM allocation: Joint clustering and non-linear dynamic modeling of sequence data. Proceedings of the 34th international conference on machine learning70. Proceedings of the 34th international conference on machine learning* International Convention Centre, Sydney, Australia: PMLR3967–3976 Proceedings of Machine Learning Research.

Zhao, J., Lan, M., & Tian, J. F. (2015). *Using traditional similarity measurements and word embedding for semantic textual similarity estimation. 9th international workshop on semantic evaluation (SemEval 2015)*117.