Bundles: A Framework to Optimise Topic Analysis in Real-Time Chat Discourse.

Jonathan Dunne*, David Malone†
Hamilton Institute, Maynooth University
Ireland
Email: *jonathan.dunne.2015@mumail.ie, †david.malone@mu.ie

Andrew Penrose
IBM Technology Campus
Ireland
Email: apenrose@ie.ibm.com

***Abstract*—Collaborative chat tools and large text corpora are ubiquitous in today's world of real-time communication. As micro teams and start-ups adopt such tools, there is a need to understand the meaning (even at a high level) of chat conversations within collaborative teams. In this study, we propose a technique to segment chat conversations to increase the number of words available (19% on average) for text mining purposes. Using an open source dataset, we answer the question of whether having more words available for text mining can produce more useful information to the end user. Our technique can help micro-teams and start-ups with limited resources to efficiently model their conversations to afford a higher degree of readability and comprehension.**

I. Introduction

We live in an information age, and consumer-based services and applications generate more text-based data. As we embrace both collaborative and social communication, we converse more often via text-based communication [1] [2]. For both business and recreational purposes real-time chat discourse appears to be part and parcel of our lives.

However, for businesses irrespective of size, using such collaborative and social means of communication, can be an overwhelming experience [3]. This is due in part to the large volumes of text-based data that are generated by such applications and services. Recent studies have shown that almost 350,000 tweets are created every minute of every day. Globally 2.5 quintillion bytes of data are produced [4]. The growth in social media messaging is not confined to tweet messages. A recent study [5] by the Harvard business school has shown that over 2.5 billion users communicate with at least one messaging app (e.g. WhatsApp, Facebook). This figure will rise to 3.6 billion users in the next few years. Therefore, for this study, we consider techniques that may help teams make sense of their message based data.

Topic modelling is a frequently used process to discover semantic structure within a text corpus. Topic modelling and text mining are used across multiple disciplines [6] as a vehicle to grow business insights [7]. For example, if a business can mine customer feedback on a particular product or service this information may prove valuable [8]. One of the recommendations when employing text mining/topic modelling techniques is that the more data available for analysis, the better the overall results. However, even in the age of big data, practitioners may have a requirement to text mine a single conversation or small text corpus to infer meaning.

In this paper, we propose a framework that both micro teams and SMEs can use to deliver a significant level of topic modelling terms, from real-time chat discourse data, while utilising their limited resource cohort. The core idea of this framework is for topic mining practitioners to partition their conversations using a novel technique. Such a method can provide a higher number of words (19% on average) for topic summarisation tooling. For small teams with a limited pool of test resources, leveraging such segmentation techniques can provide not only more words for text mining but an improved level of readability than using an entire message corpus alone.

This paper contains research conducted on an open-source real-time chat corpus. Through the study of this dataset we investigate a) If by partitioning messages based on their inter-arrival time, can a more significant number of distinct words be returned for use by topic modelling software? b) Does a higher number of words provide a level of readability that is easier for humans to comprehend? c) Can we use the results of this work to predict an optimal topic cluster size? Using the results of this study for our framework, a topic mining solution can be developed to provide an enhanced level of understanding for small message corpora.

The rest of the paper is structured in five sections: Section II gives some description of study background and related works. Section III describes the enterprise dataset. Section IV provides analysis and methodology. It is followed by section V that explains the result. Finally, the conclusion and future work are described in section VI.

II. Background and related research

We begin our review of the background literature and relevant studies, first with an overview of Natural Language Processing (NLP). Following on we provide an overview of Corpus Linguistics. We then review a number of popular tools for Topic Modelling and briefly discuss Linear regression. Finally we wrap up this section, with a review of relevant studies specifically related to small text corpora.

A. Natural Language Processing

Tokenisation is a process of converting a sequence of characters (e.g. message discourse) into a series of tokens (strings with an assigned meaning) [9]. Therefore, before any analysis is conducted on a text corpus, the text is divided into linguistic elements such as words, punctuation, numbers and alpha-numerics [10].

Stop words are words which are filtered out before or after processing of text discourse [11]. Stop words typically refer to the most common words in a language; there is no consensus or master list of agreed stop words. The website "ranks.nl" provides lists of stop words in forty languages [12]. Hans Luhn, one of the pioneers in the field of information retrieval, is credited with creating the concept of stop words [13].

Stemming is a method of collapsing inflected words to their base or root form [14]. For example, the words: fishing, fished and fisher, could be reduced to their root fish. The benefit of stemming can be seen as follows: If one is interested in term frequency, it may be easiest to merely count the occurrences of the word fish rather than its non-stemmed counterparts.

Lemmatisation is the process of grouping together the inflected words, for analysis as a single entity [15]. On the surface this process may look like the opposite of stemming; however, the main difference is that stemming is unaware of the context of the words and thus, cannot differentiate between words that have other meanings depending on context. For example, the word "worse" has "bad" as its lemma. This link is missed by stemming as a dictionary lookup is needed. Whereas, the word "talk" is the root of "talking". This reference is matched in both stemming and lemmatisation.

B. Corpus Linguistics

Corpus linguistics is the study of language as expressed in corpora (i.e. collections) of "actual use" text. The core idea is that analysis of expression is best conducted within its natural usage. By collecting samples of writing, researchers can understand how individuals converse with each other. One of the most influential studies in this field was conducted by Kučera and Francis [16]. The authors analysed an American English Corpus, that involved analysis techniques from linguistics, psychology and statistics.

C. Topic Modelling Tools

Latent Semantic Analysis (LSA) is a method that allows for a low-dimension representation of documents and words. By constructing a document-term matrix, and using matrix algebra, one can infer document similarity (product of row vectors) and word similarity (product of column vectors). The idea was first proposed by Landauer et al. in 1998 [17].

In 1999 Hofman proposed a statistical technique of two-mode and co-occurrence data [18]. In essence, his Probabilistic Latent Semantic Analysis model (PLSA), allowed a higher degree of precision for information retrieval than standard LSA models. This is due to the introduction of a novel Tempered Expectation Maximisation technique that used a probabilistic method rather than matrices for fitting. However, one drawback of the PLSA method, is that, as the number words and documents increase, so does the level of overfitting.

Latent Dirichlet allocation (LDA) is a generative statistical model that allows topics within a text corpus to be represented as a collection of terms [19]. At its core, LDA is a three-level hierarchal Bayesian model, in which each item in an array is modelled as a finite mixture over an underlying set of topics. Blei et al. first proposed the idea in 2003.

D. Linear Regression

Linear regression is a statistical technique to model the relationship between two or more variables. Typically, one or more explanatory (or independent) variables expressed as X, are used to predict the a response (or dependent) variable expressed as y. Where one independent variable is used, the process is known as simple linear regression. Where more than one independent variable is used the process is known as multiple linear regression.

At a high level, both sets of variables are plotted in the form of a scatter plot, and a least squares line is fitted between the points on the graph. This approach attempts to fit a straight line between the points plotted. If the two sets of variables have a linear relationship, a suitable linear functional can be obtained in the following form:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\epsilon}_i \tag{1}$$

Linear regression was first proposed by Francis Galton is 1886 [20].

E. Studies Related to Topic Modelling of Small Text Corpora

Jivani conducts a comparative study of eleven stemmer's, to compare their advantages and limitations [21]. The study found that there is a lot of similarity regarding performance between the various stemming algorithms. Additionally, a rule-based approach may provide the correct output for all cases, as the stems generated may not always be accurate words. For linguistic stemmers their output is highly dependent on the lexicon used, and words outside of the lexicon are not stemmed correctly.

Naveed et al. [22] investigates the problem of document sparsity in topic mining in the realm of micro-blogs. Their study found that ignoring length normalisation improves retrieval results. By introducing an "interestingness" (level of re-tweets) quality measurement also improves retrieval performance.

The Biterm topic model is explicitly designed for small text corpora such as instant messages and tweet discourse [23]. Conventional topic models such as LDA implicitly capture the document-level word co-occurrence patterns to reveal topics, and thus suffer from the severe data sparsity in short documents. With these problems identified, Yan et al., proposed a topic model that a) explicitly models word co-occurrence patterns and b) uses the aggregated patterns in the whole corpus for learning topics to solve the problem of sparse word co-occurrence patterns at document-level.

Yin et al. [24] discuss the problem of topic modelling short text corpora such as tweets and social media messages. The core challenges are due to sparse, high-dimensional and

TABLE I
SUMMARY OF DATASET CONVERSATION METRICS

Metric	Chat 1	Chat 2	Chat 3
Total Messages	46	70	59
Total Words	292	436	484
Non-Stopped Words	158	239	262
Distinct Non-Stopped words	111	168	186
% Words for analysis	38	39	38

large volume characteristics. The authors proposed a Gibbs Sampling algorithm for the Dirichlet model (GSDMM). The authors demonstrated that a sparsity model could achieve better performance than either K-means clustering or a Dirichlet Process Mixture Model for Document Clustering with Feature Partition.

Sridhar [25] presents an unsupervised topic model for short texts using a Gaussian mixture model. His model uses a vector space model that overcomes the issue of word sparsity. The author demonstrates the efficacy of this model compared to LDA using tweet message data.

Topic Modeling of Short Texts: A Pseudo-Document View by Zuo et al. [26] propose a probabilistic model called Pseudo-document-based topic model (PTM) for short text topic modelling. PTM introduces the idea of a pseudo-document to implicitly aggregate short texts against data sparsity. By modelling these pseudo-documents rather than short texts, a higher degree of performance is achieved. An additional sparsity enhancement is proposed that removes undesirable correlations between pseudo-documents and latent topics.

Schofield and Mimmo [27] investigate the effects of stemmers on topic models. Their research concluded that stemming does not help in controlling the size of vocabulary for topic modelling algorithms like LDA, and may reduce the predictive likelihood. The authors suggest that post-stemming may exploit nuances specific to corpora and computationally more efficient due to the smaller list of words for input.

III. DATA SET

Topic modelling and text mining of social media/collaboration messaging have been shown to provide insight into the subjects people discuss as part of their online communication. By segmenting instant message text in a novel way, before topic modelling, we demonstrate how a higher degree of understanding can be achieved by the results of topic model outputs.

For this study, we topic modelled three complete chat conservations from an open source Ubuntu developer IRC channel [28]. For each conversation, IRC messages were read, we noted an initial salutation, a valediction and a grouped topic discussed in-between the greeting and farewell messages. For this study, only conversations with related topic content were considered. We note that chat conversations with mixed chat messages (i.e. 'entangled chat conversations') are beyond the scope of this study and will not be considered here.

TABLE II
SUMMARY OF DIFFERENCES BETWEEN QUESTIONNAIRE SAMPLES

Sample 1	Entire chat - Topic Modelled
Sample 2	Burst & Reflections - Topic Modelled
Sample 3	Entire chat - Stop words removed
Sample 4	Entire chat - No text pre-processing

Table I provides a summary of the number of total words, the non-stopped words, the distinct non-stopped words and the percentage of words available for analysis.

This study aims to answer the following questions. First, can we segment a chat conversation in such a way as to provide a greater number of distinct words for topic modelling algorithms? Second, if a reasonable segmentation method can be found, is the output from a topic model easier to infer meaning, then modelling the entire conversation alone? Third, is there a relationship between the topic modelling cluster size and the number of words Input/Output from topic modelling?

A. Conversation segmentation

A question for practitioners of topic modelling is, how can we maximise the number of words for analysis? We know from prior research that text mining algorithms that some form of text pre-processing is required prior to topic modelling. Pre-processing may include at least one of the following: Tokenisation, stop words removal, stemming and lemmatisation. The removal of words as part of this pre-processing step usually is not an issue for a large text corpus, due to the number of words available. In the case of small text corpora, the problem may be more acute. For our study, stemming and lemmatisation was not conducted.

We recorded the inter-arrival time of instant message posts within the Ubuntu IRC channel, and grouped messages by short and long inter-arrival times. For successive messages with a zero minute inter-arrival time, we define this collection of messages as a burst. For messages with a one minute or greater inter-arrival time, we define this group of messages as a reflection. We then perform text mining on each burst and reflection period and then aggregate the output terms. For topic text mining, we used the tool Biterm, which is suited to modelling small text corpora.

B. Topic modelling comprehension

After a conversation has been a) segmented into burst and reflection periods, b) these periods topic modelled and c) the results aggregated, we consider the efficacy of the output.

We accept that the terms output from a topic model algorithm is not explicitly designed for a readable summary. Instead, they are designed to give a user an indication of the terms used in a text corpus. Nevertheless of interest is how a user can understand the output of text mining. Our approach is to prepare four sets of text as follows; 1) each conversation is modelled with Biterm (as a whole) and the mined terms output into a single collection, 2) the bursts and reflections from each conversation are modelled individually, the terms are then aggregated into a single collection, 3) each conversation with the stop words removed and 4) the raw conversation (i.e. without any pre-processing). Table II summarises the level of pre-processing conducted for each sample.

We then asked twenty four test subjects to summarise each of the four text sets belonging to a single conversation. Additionally, we asked each participant to comment on which of the four text sets was easiest to summarise. Next, we asked each subject, whether they felt set one (all terms topic modelled) or set two (bursts and reflections topic modelled) was most natural to summarise. Finally, we asked each subject to describe why they felt the text set chosen in the second question was easiest to summarise. Results of a meta-study on sample sizes for qualitative studies [29] show there is variability in sample size depending on the subject domain. For our questionnaire, twenty-four individuals were selected, and each conversation was randomly distributed among the users.

Finally we compared the readability of every text set for each conversation using a number of known readability tests such as; Dale-Chall [30], Coleman-Liau [31], Flesch-Kincaid [32] and Gunning Fog [33].

C. Term cluster size prediction

Topic modelling algorithms use a unique set of words from a corpus for analysis. Also, we know that the process of text mining may be, in part non-deterministic. In other words, random sampling is often used to generate a term list. One of the goals of text mining is to ensure that a sufficient number of words are output in each topic cluster. The intuition is that the more unique words that are provided, perhaps the easier the output will be to understand.

Biterm, outputs topic mined terms as ‘topic clusters’. Each cluster has a maximum size of ten terms. If one hundred words are input for analysis, the intuition is that ten clusters will be output with a ten distinct words. However, due to the underlying random nature of the sampling algorithm used, this is not always the case. Therefore, it is necessary to use a range of cluster sizes to obtain the optimal number of terms. We define ‘optimal output words’ as the number of words that closely matches the number of words used for Biterm analysis. We define the ‘optimal # clusters’, as the smallest number of clusters that contains the optimal output words.

Linear regression is a method to determine the relationship between two or more variables where one variable is dependent, and the additional variables are independent. A hypothesis test (are two or variables correlated) is conducted, and a p-value is computed. Depending on the size of the p-value, the hypothesis of a relationship/no relationship can be accepted or rejected. We used the lm function found in the base R package [34] and performed a linear regression test.

We will use linear regression to explore the relationship between the number of unique words input, the Biterm cluster size and the unique terms output. For example, if we model the unique words input to Biterm, the cluster size that provides the unique optimal set of terms and the count of these text mined terms, a linear model could be used to predict optimal term cluster size.

TABLE III
SUMMARY OF TEXT MINING ANALYSIS: ENTIRE CONVERSATIONS VS BURST AND REFLECTIONS

Metric	Chat 1	Chat 2	Chat 3
Total words	292	436	484
Non-stopped words	158	239	262
Distinct non-stopped words	111	168	186
Distinct non-stopped terms output	96	129	143
# Words not analysed	196	307	341
% Words for analysis	38	39	38
% Actual terms output	33	30	30
Total burst words	185	226	287
Non-stopped burst words	98	143	163
Distinct non-stopped burst words	91	118	154
Distinct non-stopped terms output	87	118	145
# Burst words not analysed	94	108	142
# Bursts	7	11	8
% Words for analysis	49	52	54
% Actual terms output	47	52	51
Total Reflection words	107	210	197
Non-stopped reflection words	61	99	99
Distinct non-stopped reflection words	60	95	94
Distinct non-stopped terms output	60	93	94
# Reflection words not analysed	47	115	103
# Reflections	7	12	9
% Words for analysis	56	45	48
% Actual terms output	56	44	48

D. Limitations of dataset

The dataset has some practical limitations, which are now discussed. The process of aggregating chat messages into a cohesive conversation is a subjective one. Every effort was made to assign messages to their most appropriate thread. We recognise that the process is subjective. Additionally, the post times for the Ubuntu chat were measured in hours and minutes only. As a result, we defined our burst and reflection period as timed in zero and one minute or greater priors respectively.

The chat conversations that form part of this study are from an Ubuntu IRC developer channel. While we hope these examples will be representative of technical discussion channels, it seems unlikely they will be typical of all types of channels.

IV. RESULTS

We now explore the results of our analysis.

A. Conversation segmentation

Table III shows a summary of the topic modelling conducted on each of the three conversations. In the first experiment, the entire discussion was mined. In the second experiment, the burst and reflections were modelled separately.

For conversation one, a total of 96 terms were output by Biterm when modelling the entire text, whereas 87 and 60 terms respectively were output from the burst and reflection

TABLE IV
SUMMARY OF TEXT SAMPLE QUESTIONNAIRE ANSWERS (Q1 & Q2)

Question	**Sample 1 - Biterm (All text)**	**Sample 2 - Biterm (Burst & Reflections)**	**Sample 3 - (Stop words removed)**	**Sample 4 - (Full text)**
One: Of the 4 text samples, which sample did you find easier to summarise? (1/2/3 or 4)	0	0	2	22
Two: Of samples 1 and 2, which sample did you find easier to summarise? (1 or 2)	0	24	NA	NA

analysis. A total of 51 (17%) more terms were output from the combined burst and reflection analysis than modelling the entire conversation.

For conversation two, a total of 129 terms were output by Biterm, whereas 118 and 93 terms respectively were output from the burst and reflection analysis. A total of 82 (19%) more terms were output from the combined burst and reflection analysis than modelling the entire conversation.

For conversation three, a total of 143 terms were output by Biterm, whereas 145 and 94 terms respectively were output from the burst and reflection topic mining. A total of 96 (20%) more terms were output from the combined burst and reflection analysis than modelling the entire conversation.

B. Topic modelling comprehension

Recalling the survey questions asked in section III part B: Of the four text samples, which sample did you find easier to summarise? And of samples 1 and 2, which sample did you find easier to summarise? Table IV shows a summary of the answers to the questions asked of the test subjects. Before the questionnaire, the subjects were asked to summarise the four samples. The questions were asked directly after the summarisation task. As we can see for question one, the majority of users found sample 4 easiest to summarise. For question two, the respondents answered unanimously in favour of sample 2.

Question three asked: For the sample, you chose in question two, why did you find that text sample easier to summarise? Fig. 1 shows a word cloud generated from the answers respondents provided. When stop words were removed, the following terms appeared most frequently: easier (8 times), text (6), words (5) and flow/natural/understand/words (all 5).

To further understand the readability of text output from our topic mining experiments, we conducted some readability tests (Dale-Chall, Coleman-Liau, Flesch-Kincaid and Gunning Fog) against each of the four text samples for all three

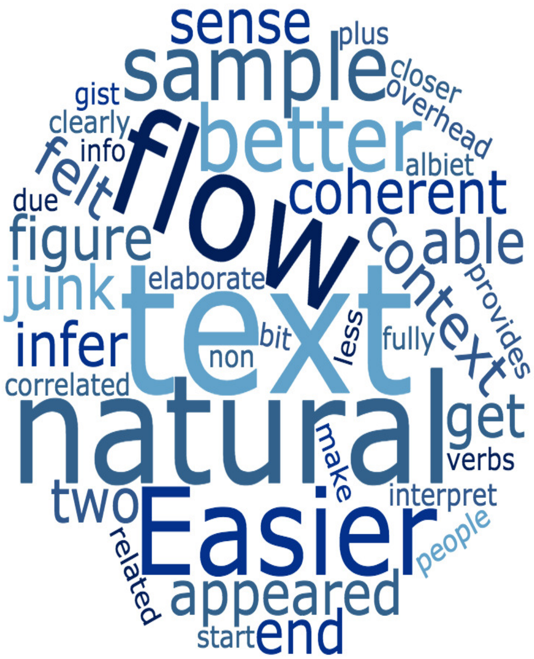

Fig. 1. Word Cloud of answers from Question 3

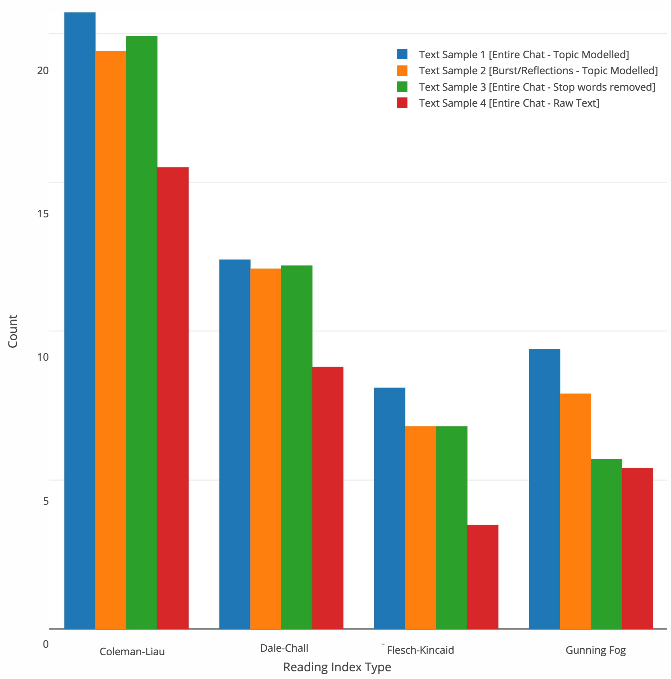

Fig. 2. Mean readability index score of each text sample

TABLE V
LINEAR REGRESSION COEFFICIENT TABLE

Coefficients	**Estimate**	**Std. Error**	**t value**	**Pr(>t)**
(Intercept)	0.934	0.122	7.606	3.45e-10
Optimal.Terms	0.058	0.003	18.201	<2e-16

conversations. Fig. 2 shows a bar chart of mean readability index scores. In all cases, a lower score indicates a more readable text sample. Intuitively we can see that text sample 1 had the highest score across all indices, and text sample 4 had a lowest. Text sample 2 had a lower index score than sample 1 in all readability tests.

C. Term cluster size prediction

Our third research question asked, "Can we use the results of our topic modelling to predict an optimal topic bundle size?" We mentioned previously that obtaining an optimal number of terms (i.e. an output number of distinct words that matches an input number of distinct words) from Biterm is an iterative process.

For each burst, reflection and complete segment we topic modelled multiple cluster numbers to obtain the optimal number of distinct words. Once an optimal cluster size was found, the number of clusters was noted. We then used Linear regression to determine if there is a strong relationship between the number of distinct words output and the cluster size. In order words can we create a linear function to predict the number of topic clusters, if the optimal number of terms are known?

Table V shows the output of the Linear regression analysis. We also note the following additional outputs; Residual standard error: 0.818, Multiple R-squared: 0.855, Adjusted R-squared: 0.853 and p-value: <2.2e-16. From the output we can see that the equation to fit our linear model is as follows:

$$NumberofClusters = 0.934 + 0.058(Optimal.Terms) \tag{2}$$

Figure 3 shows the four goodness-of-fit plots generated from our linear regression model. These plots are used in conjunction with the results of the Linear coefficients table to determine the suitability of the model. We shall discuss these results of the model in more detail in the next section.

V. DISCUSSION

The following section provides deeper analysis and discussion of the results. In each section, references will be made to each research question asked in section III.

A. Conversation segmentation

Our first research question asked, can we segment a chat conversation in such a way as to provide a higher number of unique words for topic modelling algorithms? Table III shows that the mean proportion of words available for analysis for topic modelling of an entire conversation is 38%, this is due to the considerable number of stop words that are removed as part of pre-processing. Likewise, the mean number of terms output from Biterm is 31% a reduction of 7%.

Conversely, when both burst and reflections are aggregated, the mean proportion of terms available for analysis is 51%. Furthermore, the mean proportion of terms output from Biterm is on average, 50% a reduction of only 1%.

There is evidence to suggest that segmenting conversations into shorter segments provides a greater number of words for topic analysis due to the lack of duplicate words found in each smaller segment. We note that stop words are removed irrespective of the segment size.

Some interesting points are raised by our analysis. When stop words are removed, duplicate occurrences of the same word are also discarded. However, in the case of a more substantial text corpus, some duplicate non-stop words remain, these words are ignored by text mining tools. We see this is not the case with burst and reflection text segments. In fact, for conversation 2, 143 non-stopped words were retained. A further 25 non-stop duplicate words were ignored. In all other cases, the number of duplicate words ignored by Biterm after stop words were removed was less than 10.

Furthermore, we observed that the number of burst and reflections created might have little significance on the number of terms output. Conversations 1 & 3 had a similar number of segments (i.e. between 7 and 9), while conversation 2 had 11 burst and 12 reflections respectively. While no formal correlation tests were conducted, when we look at the segment size and the % terms output, there seems little positive or negative relationship between the two variables.

B. Topic modelling comprehension

Our second research question asked: If a reasonable segmentation method can be found, is the output from a topic model easier to understand, than modelling the entire conversation alone? In other words, even if more words can be output as part of our improved segmentation technique, how does this translate into comprehension by both a human and for general readability.

Table IV summarises the answers to the first two questions asked by the 24 individuals who took part in our topic modelling comprehension experiment. Unsurprisingly we can see that the majority of respondents picked sample 4, as the easiest to summarise. The consensus was that with all words available and with grammar respected (to a degree) sample 4 was easiest to summarise for the majority. However, we note that two respondents picked sample 3 (stop words removed). The feedback from these two participants was that the samples with fewer words were easier to understand, this may be because these two individuals were not Ubuntu experts.

For question 2 the unanimous feedback from all users was that sample 2 was much easier to read than sample 1. A word cloud produced from the short answers provided, clearly indicate that a combination of our segmentation technique and Biterm preserved the natural flow of the conversation to the

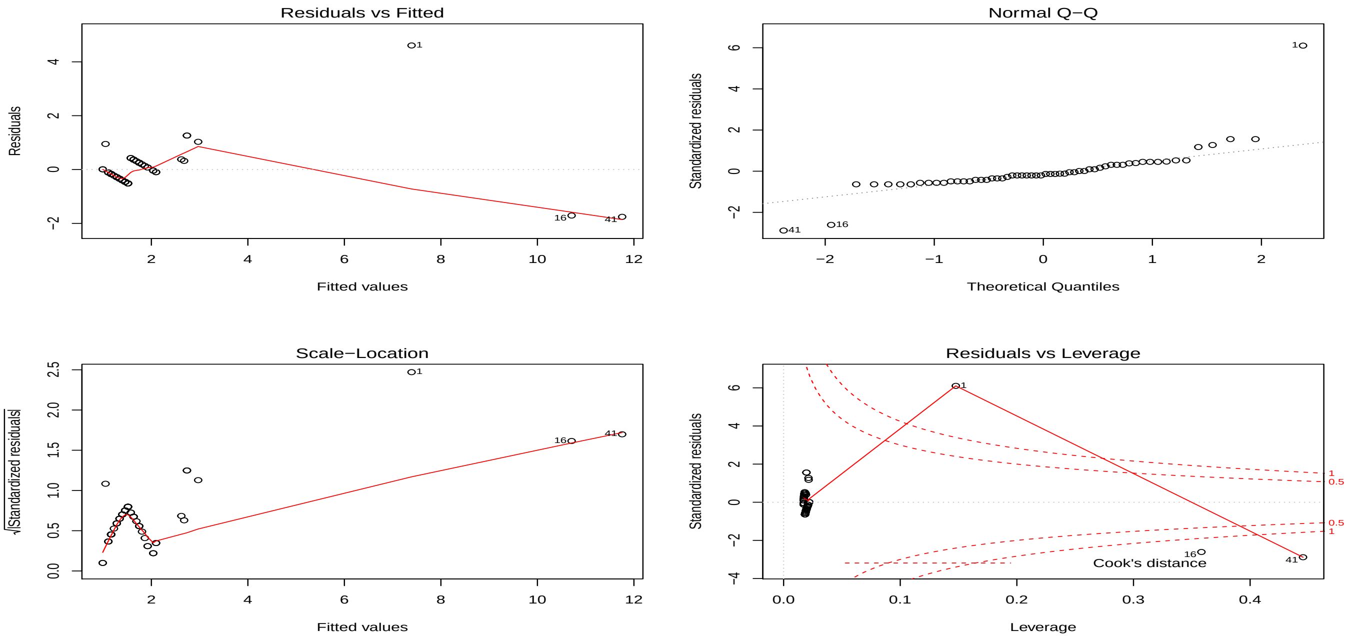

Fig. 3. Residuals Vs Fitted, Q-Q Plot, Scale-Location & Residuals Vs Leverage

degree that it was easier to summarise the text sample than sample 1.

Turning to the readability tests conducted, we can see that sample 4 produced the lowest mean index score indicating that the unprocessed chat was the most readable based on the four tests conducted. Except for the Gunning Fog index, sample 2 had equal or lower readability scores than sample 3.

It would be over-simplistic to state that when more words are available, it is easier for a human to understand a text segment based on a list of topic terms. However, it seems reasonable to assert that when more words are available and when words are placed in a similar order as to how they were typed, it is easier for humans to comprehend. What was interesting to note for short burst and reflection segments, (i.e. ten words or less) the input order of words was the same as the output terms produced by Biterm. That is to say the word at the start of the sentence had the highest log-likelihood value, while the word at the end of the sentence had the lowest log-likelihood value.

We note that the goal of this research question was not to provide a readable summary based on text mined terms. Instead, the goal was to assess the understanding of text samples to humans when varying degrees of topic mining is conducted.

C. Term cluster size prediction

Our third research question asked: Can we use the results of our topic modelling experiments to predict the optimal cluster size? Previously, we discussed the problem of determining the number of clusters that will return the highest number of distinct words from the Biterm analysis. We also mentioned that the optimal cluster number could be obtained only by iteratively trying a range of sizes.

Table V provides the output of a linear regression experiment whereby we used the optimal terms to predict cluster size. The first point to note is that the p-value for optimal.terms was <2e-16, this figure indicts a strong regression effect. Additionally, we note that the multiple R-squared and adjusted R-squared values were 0.855 and 0.853. These values indicate the model is an excellent fit for our data.

Fig. 3 depicts four goodness-of-fit plots to assess the goodness of fit of our model graphically. The residuals Vs fitted plot shows our model passes through the majority of fitted values quite well. It appears that a small number of points are outside the fitted line. The normal Q-Q plot shows the standardised residuals fitted against a normal distribution line. For the most part, almost all values fit the line. There are a number of exceptions, such as a few small and large residuals. This plot indicates the model would almost all values however there would be some uncertainty of fitted very small and very large values. Finally, three observations had leverage greater than 0.1.

We mentioned previously that Biterm topic clusters contain ten terms, and that due to random variation of the tooling, it is not always possible to obtain the same level of distinct output terms as were input. For example, dividing the number of distinct words for analysis by ten and using this result as the optimal # of clusters, does not provide the same amount of output words. However, we did not know to what degree of fit a linear model would provide. In our case, a good fit was obtained.

The main benefit such a linear model is as follows: Determining the optimal cluster size can be a time-consuming task, especially for large datasets. Even using a rule of thumb such as a 'divide input distinct words by ten' as a starting point, multiple iterations of Biterm may be required. By using a linear model such as ours, the task of determining the optimal term cluster size may be expedited.

VI. Conclusion

The purpose of this study was to examine topic modelling for small text corpora (i.e. Instant message conversations). We found that by segmenting messages into periods of intense (bursts) and non-intense (reflections) communication that these segments, when used in conjunction with a text mining tool can be used to provide a higher number of output terms than modelling the entire corpus of messages at once. Furthermore, we found that the message inter-arrival time can be used to determine both burst and reflection periods.

We also found that the terms output from topic modelling bursts and reflection periods, when aggregated, is easier to understand than the text mined terms from the entire message corpus. Additionally, we saw that all four readability tests, topic modelled terms output from aggregated burst and reflection analysis have a lower readability index compared to terms mined from the entire corpus.

Finally, the relationship between optimal output words and the optimal # clusters had a strong regression effect. In other words, we can use the optimal terms to predict the required number of topic clusters. This result can have a positive benefit for topic modelling practitioners, as it may reduce the iterative approach needed to find the number of topic clusters that produce the largest distinct number of words.

Both SMEs and micro-teams can use the above result to deliver high-value topic mining outputs from their group chat discourse. Teams can focus initially on a corpus-based approach for a particular channel/space. The advantage of a more extensive corpus approach is that topic modelled outputs can be assessed in context. Where words collations exist, this knowledge can be directly applied to place a higher value on terms generated from topic mining tools.

In future work, we shall model burst and reflection period's on a corpus basis to infer the optimal duration.

VII. Acknlowdgements

The authors would like to personally thank the 24 individuals who took part in our topic modelling comprehension experiment.

References

[1] (2015) We just don't speak anymore. [Online]. Available: http://bit.ly/2yDXzJ6

[2] (2015) 73 Texting Statistics. [Online]. Available: http://bit.ly/2kjHeF8

[3] (2016) How to Deal With Social Media Overwhelm. [Online]. Available: http://bit.ly/2yN5e8r

[4] (2016) Expect more chatbots. [Online]. Available: http://bit.ly/2z771cJ

[5] (2017) Social Messaging: Catalysing The Next Wave of Digital Revolution In Communication. [Online]. Available: http://bit.ly/2FekIpz

[6] (2017) The Value and Benefits of Text Mining. [Online]. Available: http://bit.ly/2zJcDcl

[7] (2017) Gain business insight with Big Data. [Online]. Available: http://bit.ly/2zPxmcC

[8] (2015) Improving the Consumer E-commerce Experience Through Text Mining. [Online]. Available: http://bit.ly/2z8eYyv

[9] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson London, 2014, vol. 3.

[10] J. J. Webster and C. Kit, "Tokenization as the initial phase in NLP," in *Proceedings of the 14th conference on Computational linguistics-Volume 4*. Association for Computational Linguistics, 1992, pp. 1106–1110.

[11] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.

[12] (2017) Stopword Lists. [Online]. Available: http://bit.ly/2jwKvDa

[13] H. P. Luhn, "Key word-in-context index for technical literature (kwic index)," *Journal of the Association for Information Science and Technology*, vol. 11, no. 4, pp. 288–295, 1960.

[14] J. B. Lovins, "Development of a stemming algorithm," *Mech. Translat. & Comp. Linguistics*, vol. 11, no. 1-2, pp. 22–31, 1968.

[15] D. Manning, "Introduction," in *Introduction to Industrial Minerals*. Springer, 1995, pp. 1–16.

[16] H. Kučera and W.-N. Francis, *Computational analysis of present-day American English*. Dartmouth Publishing Group, 1967.

[17] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[18] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[20] F. Galton, "Regression towards mediocrity in hereditary stature." *The Journal of the Anthropological Institute of Great Britain and Ireland*, vol. 15, pp. 246–263, 1886.

[21] A. G. Jivani *et al.*, "A comparative study of stemming algorithms," *Int. J. Comp. Tech. Appl*, vol. 2, no. 6, pp. 1930–1938, 2011.

[22] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi, "Searching microblogs: coping with sparsity and document quality," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 183–188.

[23] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 1445–1456.

[24] J. Yin and J. Wang, "A dirichlet multinomial mixture model-based approach for short text clustering," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 233–242.

[25] V. K. R. Sridhar, "Unsupervised topic modeling for short texts using distributed representations of words." in *VS@ HLT-NAACL*, 2015, pp. 192–200.

[26] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, and H. Xiong, "Topic modeling of short texts: A pseudo-document view," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2105–2114.

[27] A. Schofield and D. Mimno, "Comparing apples to apple: The effects of stemmers on topic models," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 287–300, 2016.

[28] (2017) Ubuntu IRC Logs. [Online]. Available: https://irclogs.ubuntu.com/

[29] (2017) Qualitative Sample Size. [Online]. Available: http://bit.ly/2hWeh3R

[30] E. Dale and J. S. Chall, "A formula for predicting readability: Instructions," *Educational research bulletin*, pp. 37–54, 1948.

[31] M. Coleman and T. L. Liau, "A computer readability formula designed for machine scoring." *Journal of Applied Psychology*, vol. 60, no. 2, p. 283, 1975.

[32] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," Naval Technical Training Command Millington TN Research Branch, Tech. Rep., 1975.

[33] R. Gunning, "The technique of clear writing," 1952.

[34] Fitting linear models. [Online]. Available: http://bit.ly/2dvqYet