
Proceedings Paper:
Ringer, Charles, Nicolaou, Mihalis and Walker, James Alfred (Accepted: 2020) TwitchChat: A Dataset for Exploring Livestream Chat. In: THE 16TH AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE AND INTERACTIVE DIGITAL ENTERTAINMENT (AIIDE). AAAI Press. (In Press)

TwitchChat: A Dataset for Exploring Livestream Chat

Charles Ringer*, Mihalis A. Nicolaou‡, James Alfred Walker*
*University of York, United Kingdom
‡The Cyprus Institute, Cyprus
cr1116@york.ac.uk, m.nicolaou@cyi.ac.cy, james.walker@york.ac.uk

Abstract

Most natural language processing research focuses on modelling and understanding text formed of complete sentences with correct spelling and grammar. However, livestream chat is drastically different. Viewers are typically writing short messages while responding to in-stream events, often with incorrect grammar and many repeated tokens. Additionally, tokens that are commonly used in livestream chat are unknown to traditional language understanding efforts that focus on prosaic text. To advance and encourage further research in terms of livestream chat understanding, in this work, we present a large-scale dataset of video game livestream chat, consisting of over 60 million tokens. As livestreaming becomes more popular it is also increasingly pertinent to study, though chat analysis, the way in which the audience is engaging with the stream. However, this is not a straightforward task, livestream chat is a rich and complex domain, far removed from often studied prosaic text. Additionally. we provide a case study analysis of word vector methods applied to the dataset, showing that the vector space is strangely shaped but clusterable and that the resulting clusters correlate with features such as streamer popularity. Furthermore, human relatedness tests highlight the difference that this domain poses with respect to prosaic text. It is hoped the livestream chat dataset, the discussion of its unique features, and the challenges highlighted for future work will invigorate the research community into further study of livestream chat.

Introduction

Livestream chat, e.g. from Twitch.tv, is a particularly interesting language domain to study, the information in the chat may be an indicator of events in the livestream (Jiang et al. 2020). However, it is a unique and complicated domain. It contains many misspellings, often due to viewers speedily typing messages reacting to the stream. These misspellings can also be intentional e.g. 'leet' speak and other spelling modifications (Blashki and Nichol 2005). Additionally, chats make heavy use of 'emotes' (Barbieri et al. 2017b), domain-specific emoji with rich, complicated meanings. A scrolling window of chat messages, as shown in Figure 1, is presented at the side of the livestream where viewers can see several previously sent messages. Messages are displayed until enough new messages have been sent that the message leaves the chat window. The higher the volume of messages being sent the less time a message is visible.

While livestream chat has some similarities with other social media data, it is strongly evinced in the literature that livestream chat has several unique properties. Primarily, it is strongly linked to the streamed game content (Recktenwald 2017; Musabirov, Bulygin, and Okopny 2017), e.g. because participants react to events in the stream, and is thus fundamental in understanding the context of livestreams. Secondly, the huge scale of the chat size in addition to time constraints results in unique properties. Particularly, most tokens are completely unknown to existing social media focused lexicons e.g. Vader (Hutto and Gilbert 2015). Understanding this domain is, therefore, both extremely challenging and also very important as we see the rise in popularity of live streaming as an entertainment domain.

In this work, we present a large scale data-set[1] of chat text. Additionally, we present a baseline investigation of this dataset which highlights how unique and challenging the domain is. To do so we utilise Skip-gram Negative Sampling (SGNS) (Mikolov et al. 2013b) using human model evaluation. SGNS is known to produce curious vector spaces (Mimno and Thompson 2017) but, in spite of this, we find that livestream chat vector spaces have a particularly strange shape. Finally, we suggest potential areas of research utilising livestream chat data. Throughout this work the term 'token' will be used to refer to words, emotes and emojis.

Related Work

Related work can be split into three categories. Firstly, prior work into understanding and modelling livestream chat. Secondly, work into developing word vectorisation techniques, especially those focused on SGNS approaches. Thirdly, the limited study into Emoji and the heavy use of ideograms as a defining feature of livestream chat.

Twitch Chat

Prior research has focused on viewer communities (Hamilton, Garretson, and Kerne 2014; Seering, Kraut, and Dabbish 2017) as well as toxicity in these communities (Poyane 2018; Bulygin 2018). Work has also been undertaken

[1]Data, code, and extra case-study visualisations are available at https://osf.io/39ev7/

into modelling streamers and viewers through graphs and finite state machines (Nascimento and Ribeiro 2014). Additionally, we see that the majority of viewers are watching the most popular streams (Kaytoue, Silva, and Cerf 2012). An important observation is that while there are often many individual users sending messages, as shown in Figure 1, they tend to follow a set of coherent 'voices' or personas (Ford, Gardner, and Liu 2017; Cheung and Huang 2011; Smith, Obrist, and Wright 2013). This 'crowd speak' results in a coherent chat stream, even if the messages are coming from many authors. Outside of natural language processing, there has been a linguistic study of livestream chat that observed significant variations compared to other internet communities (Olejniczak 2015), and that chat content is heavily related to stream content (Recktenwald 2017; Musabirov, Bulygin, and Okopny 2017).

While these prior works begin to uncover the properties of livestream chats they do not offer a statistical or machine learning approach to understanding the meaning of tokens through analysis of the way the chat is constructed. While some prior works attempt to uncover the meaning of stream specific tokens, e.g. (Barbieri et al. 2017a), generally these works assume that certain tokens, such as 'Kappa', have a certain meaning. In contrast, our work attempts to learn semantic vector spaces through a study of the data itself. Finally Nakandala et. al. (Nakandala et al. 2017) explored the use of gendered conversation in the livestream context by utilising a set of vectorisation techniques.

Word Vectors

Word vectorisation is a popular approach for learning meaningful numeric representation of tokens (Turney and Pantel 2010)(Mikolov et al. 2013a), usable in downstream tasks, e.g. Neural Networks, (Bakarov 2018). Of particular interest to our work is research into SGNS methods (Mikolov et al. 2013b), which are particularly attractive for two reasons. Firstly, they are weight-efficient. Secondly, they require data in the format of two tokens and a label. This label is usually a $[0, 1]$ binary value, describing if the samples are close together in the corpus, a formulation which can be modified to accommodate the temporal aspect of livestream chat. Analysis of SGNS models has shown that they generate unusually shaped vector spaces (Mimno and Thompson 2017) due to the impact that the negative sampling objective has, although they do still retain the ability to encode semantics. An advantage of word vector models for livestreaming is that they do not require prior knowledge of the semantic meaning of tokens, which is useful when dealing with livestream specific words and emotes whose meanings are uncertain.

Emoji

Emoji and other ideograms are popular within livestream chats, so recent analyses of the way that Emoji are used is relevant. For instance, (Wiseman and Gould 2018) shows that ideograms often have a complicated, and often highly personal meaning. We can reasonably assume therefore that certain emoji and emotes evolve to have a rich Twitch.tv specific meaning, although the meaning of ideograms within livestream chats is not the subject of this work. More generally, work has been carried out into how ideograms are used in conversant text. For instance, we see that norms surrounding the meaning of ideograms propagate through social networks (Park et al. 2013) and that geographic location, in our case perhaps tied to the nationality of the streamer, can affect ideogram usage (Ljubešić and Fišer 2016).

Figure 1: Example screenshot of livestream chat from a *League of Legends* livestream.

Table 1: Summary statistics describing the distribution of several dataset features. IQR: Interquartile Range. 'Viewers per Stream' was recorded at the start of data gathering.

Feature	Median	IQR	Min	Max
Stream Documents per Streamer	1	2	1	21
Viewers per Stream	2,211	5,586	0	165,371
Messages per Stream Document	5,196	124,340	2	483,230
Message Length	2	4	1	266

The TwitchChat Dataset

Data Collection

This paper presents a dataset gathered from Twitch.tv between June and October of 2019 by selecting the most popular channel, motivated by (Kaytoue, Silva, and Cerf 2012), for a set of games and recording all messages being sent in that channel until it went offline, then repeating this process. Twenty different games were selected, representative of the most popular games on the platform when data gathering started. Channels were only considered if they were streaming in English, as that is the most popular language on the platform and the common language among the authors.

The process outlined above resulted in a large dataset of over 60 million tokens from 1,951 documents, where each document represents text from a single stream session. Data was gathered from 666 different streamers. Summary statistics regarding the distribution of various document features can be found in Table 1. All references to streamers and users in the dataset have been replaced using a 'salt and hash' anonymisation function.

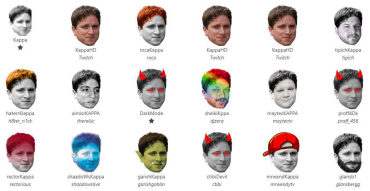


Figure 2: Variations of 'Kappa' (top left) collected using the 'View Similar Emotes' feature on https://twitchemotes.com (accessed 08-10-2019).

Table 2: Dataset size (in terms of number of tokens and number of unique tokens) before and after data cleaning stages.

Metric	Raw Dataset	After Stage 1	After Stages 2 & 3
Tokens	61,040,692	47,783,915	38,751,630
Unique Tokens	1,658,055	1,405,084	10,011

Data Cleaning Process

Token Cleaning The first stage of the cleaning process is the removal of stop words, private messages, 'bot' messages (messages sent by an automatic bot rather than a human), and generally malformed and unusual tokens, e.g. URLs. Additionally, several lemmatisation techniques were applied. Firstly, popular emotes, e.g. Kappa, often have many variants, as shown in Figure. 2, which were lemmatised so that variants are treated the same. We also see heavy use of word expansion, e.g. 'good' becomes 'goooood', as well as other common slang spellings, e.g. 'would've' becomes 'woulda'. In total, we apply 5 custom emote rules and 38 custom word expansion/misspelling rules. Lastly, generic lemmatisation using the NLTK Wordnet engine (Miller 1995; Loper and Bird 2002) was applied. This process resulted in a dataset of approx 47 million tokens.

Token Selection Once the dataset has been cleaned and lemmatised, we next select a subset of the tokens to train our models on. We focus on a subset because otherwise, the size of the model would be infeasibly large. Inspired by (Mikolov et al. 2013a), we limit our vocabulary to the 10,000 most popular tokens. These tokens are selected by assigning each token a 'document frequency' score, which represents the number of documents that the token appears in. All tokens that had a document frequency score greater than or equal to the 10,000th most frequent token were selected, resulting in a 10,011 token vocabulary. Selecting tokens based on document frequency rather than raw frequency was preferable because we are interested in gaining an understanding of livestream chat in general. Selecting tokens based on raw frequency would have resulted in certain tokens, mostly emotes, which are streamer specific and thus are very heavily used in those streams but not across Twitch in general.

Final Cleaning Finally, we remove all tokens that were not selected in the previous step. This resulted in a final dataset of 39M tokens, around 63% of the initial dataset, evidence that we can reduce the number of unique tokens and thus model size while retaining a large amount of data. Figure 3 shows a Log Rank-Log Frequency graph of all $\sim$10k tokens in the final cleaned dataset. Distributions which follow Zipf's law (Zipf 1935) have a linear relationship. However, both our most and least popular tokens do not follow this. Instead, we see that the distribution appears to be Zipfian after the $\sim$26 most popular tokens. Also, a change in gradient around rank 3,000 is observed, where tokens are used less frequently than expected with a Zipf distribution. This is an interesting observation given that most language is Zipfian, and further shows how unique livestream chat is.

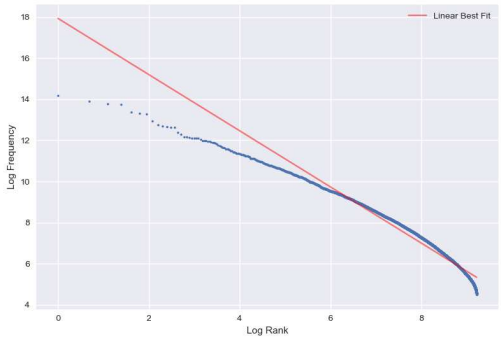


Figure 3: Log Frequency/Log Rank for the cleaned dataset. Zipfian distributions are linear when log transformed.

Case Study: Word Vector Models

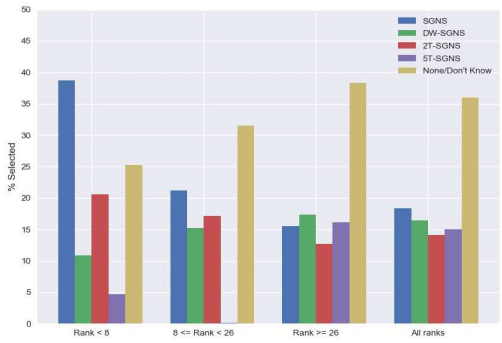
To demonstrate the TwitchChat dataset's unique features, an analysis using four word vector models is provided.

SGNS Word Vectors

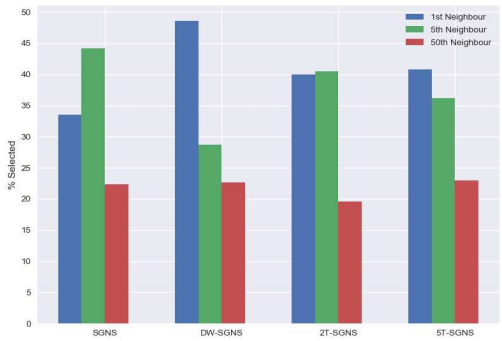
Traditional SGNS models use the spatial distance between tokens as cues for semantic similarity. Briefly, given a predefined 'window' of size l and a selected 'target' token, a second 'context' token from within this window is selected and this token pair is assigned a label of 1. Next, a 'negative' token is selected from outside of this context window, often by sampling randomly from all tokens in the vocabulary. This token pair is assigned the label 0 and acts as noise to aid learning semantic vectors(Mikolov et al. 2013b). A model is trained by finding embeddings associated with each token, from within each token pair, and then evaluating the cosine similarity of these embeddings, a $[0, 1]$ bounded function which describes the angle between two vectors. Loss is assigned based on the distance between the cosine similarity and the training label. In this way, token pairs commonly collocated in the corpus are co-located in the vector space. For more details, please see (Mikolov et al. 2013b).

Dynamic and Temporal SGNS

Developing a livestream vector model requires special attention due to several observations. Firstly, livestream chat is naturally temporal. Prosaic text is written without time constraints but chat text needs to be written promptly, e.g. reacting to in-stream events. Additionally, this causes fluctuation in the frequency of messages. Because viewers are often reacting to stream events, we hypothesise that the temporal



(a)



(b)

Figure 4: Relatedness survey results. (a) shows the % each model was selected across different frequency rank bands. (b) shows the % a neighbour was selected for each model.

distance between messages may be an indicator of the relatedness of the tokens contained within them. Additionally, we observe that livestream chat messages often have many repeated tokens and are not formed of full sentences, meaning that a fixed window size may not be appropriate. Therefore, we propose two modifications to the standard SGNS model. Firstly, we dynamically expand the context window to include any tokens within a message, which are not the target token. Secondly, rather than only sampling negative samples from noise, we sample from other messages sent in the stream, but additionally allow for some messages, if they are sufficiently close temporally, to contribute non-0 labels. In theory, sampling from real text results in negative samples, which approximate the unigram distribution whilst incorporating temporal distances. To do this we reformulate the initial binary classification task as a regression task and then derive our training labels from a transformation of the temporal distance between messages containing the paired tokens. A Gaussian Radial Basis Function (RBF) is used:

$$f(x) = ae^{-(x-b)^2 2c^2} \quad (1)$$

Where x is the temporal distance between messages in seconds, e is Euler's number, a is a constant which controls the maximum value for the function, b is another constant which controls the x value where the peak occurs, and c which is a third constant which controls the slope of the curve. For our work, $a = 1$ and $b = 0$, such that the maximum value, 1, occurs when there is 0 temporal distance between the messages. We experiment with two different values for c, $c = 2$, and $c = 5$. We use an RBF as we hypothesis that the slope of the curve is similar to the ground-truth value. Additionally, we provide a model which implements dynamic windowing but does not allow a temporally close message to contribute non-0 labels.

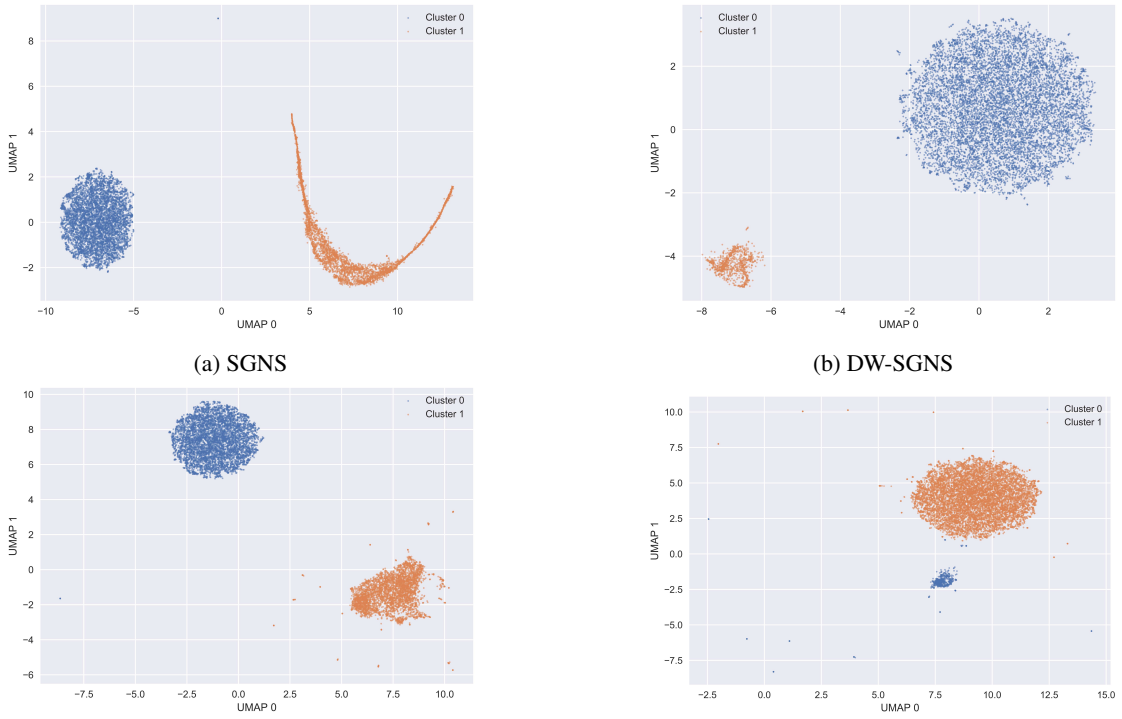
Experiment

Four word-vector models are employed. Firstly, a traditional SGNS algorithm. Secondly, a variant using dynamic windowing with l = message length (DW-SGNS). Finally, the two temporal model SGNS variants, 2T-SGNS ($c = 2$) and 5T-SGNS ($c = 5$). Inspired by (Mikolov et al. 2013a) we use an embedding dimension of 300. Each model was trained for a total of 5,000 epochs, where each epoch consists of 500 mini-batches and each minibatch contained 16,384 training samples drawn randomly from the dataset. SGNS and DW-SGNS are both trained using binary cross-entropy, whereas the temporal models are trained using Mean Squared Error. All models were implemented using Tensorflow's Keras API (Abadi et al. 2015; Chollet and others 2015). Two evaluation techniques are applied. Firstly, a relatedness test is performed which uses human evaluation of semantic embeddings. Secondly, the vector spaces themselves are explored.

Relatedness Test Details

Evaluation is difficult because livestreaming is a unique domain so traditional techniques are not suitable, e.g. test sets for livestream data do not exist. Furthermore, it is impossible to transfer livesteam models to traditional tasks because the majority of the tokens are unique to this domain. For comparison, only 13% of the tokens in our vocabulary exist in the Vader sentiment engine (Hutto and Gilbert 2015). Therefore, we must use evaluation techniques that do not require prior knowledge about the meaning of tokens.

Following from (Schnabel et al. 2015), we evaluate our models through a crowd-sourced relatedness test. To do this, we first selected the 100 most popular tokens from our dictionary, referred to as 'target' tokens. Next we queried each model and retrieved the 1^{st}, 5^{th} and 50^{th} nearest neighbours ('neighbours') for each target. Human participants were shown 20 targets alongside its neighbours from all models and are asked to select the neighbour which is most related to the target. If two competing models share a neighbour, the neighbour is only presented once. Participants were gathered through online advertising on social media sites, such as reddit.com and twitter.com. After data cleaning, where responses with no answers or only 'None/Don't Know' were removed, we had a total of 154 respondents. Because respondents were not forced to answer every question presented, there was variance in the number of responses to each question, ranging between 14 and 35 participants, with a median of 24 respondents.



(a) SGNS

(b) DW-SGNS

(c) 2T-SGNS

(d) 5T-SGNS

Figure 5: UMAP transformation of the vector space for each model (a-d). Each data-point represents a word in the vocabulary. K means clustering has also been applied to the UMAP space (K for each model was selected through the elbow method).

Results

Relatedness Test

Figure 4 (a) shows the performance for all models across 3 token rank strata, those with ranks 1 to 7, those with ranks 8 to 26, and those with rank greater than 26, motivated by Figure 3. Figure 4 (b) shows the percentage amount that each neighbour, 1^{st}, 5^{th} and 50^{th}, was selected when a model was chosen by the participants.

Visualising the Vector Spaces

The vector spaces have dimension = 300 so the shape of the space is hard to visualise. Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes et al. 2018) allows us to inspect both the global and local geometry of a vector space in fewer dimensions. Applying UMAP shows that all vector models have two core clusters, separable via k-means clustering, as shown in Figure 5.

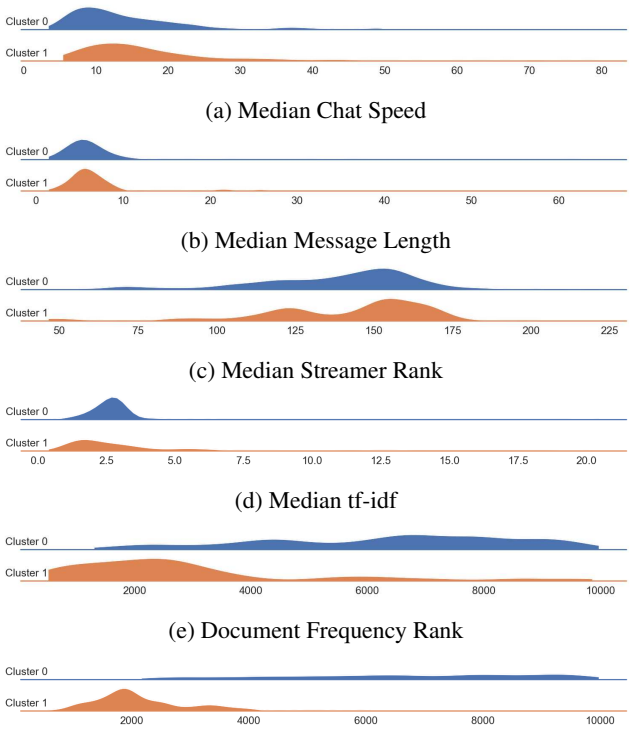
Figure 6 shows the kernel density estimation (KDE) of features for the 100 words closest to each cluster centroid. Due to space constraints, only the SGNS model is presented here, other models exhibit similar behaviour. 'Chat speed' is the number of messages sent in the 10 seconds surrounding a token's appearance. 'Message Length' is the average length of the messages containing the token. 'Streamer Rank' describes the popularity of the streamers who's stream this token appears in. Streamer Rank is calculated by ranking the mean number of views the streamer had, the most popular streamer is assigned the rank of 1. 'Term Frequency-Inverse Document Frequency' (tf-idf) describes how important a certain token is to a given document. It is calculated by multiplying the 'term frequency', the number of times the token appears in a given document, by the 'inverse document frequency', the log of the number of documents divided by the number of documents a token appears in. High tf-idf indicates tokens that appear often in a given document but not in many other documents. 'Document Frequency Rank' is the number of documents a token appears in and 'Token Frequency Rank' is the total number of appearances.

Discussion

Relatedness Test

No model outperforms the others in all situations, although SGNS appears to be generally the best. SGNS is the most selected model and is overwhelmingly the best model with very popular tokens. This is the only set of questions where more respondents chose one of our models over 'None/Don't Know'. However, for lower-ranked tokens, the performance gap is closed, with DW-SGNS and 5T-SGNS models being selected more often than SGNS for tokens with rank ≥ 26, although performance is poor for all models.

It is not clear if poor performance on lower-ranked tokens is due to poor quality models or because these tokens are less understood by participants and thus the quality of response is lower. We queried Vader (Hutto and Gilbert 2015) to see



(a) Median Chat Speed

(b) Median Message Length

(c) Median Streamer Rank

(d) Median tf-idf

(e) Document Frequency Rank

(f) Token Frequency Rank

Figure 6: KDE plots comparing the distributions of the 100 tokens closest to the centre of each cluster for SGNS.

which tokens exist in its lexicon and which are Twitch specific, finding that only 21 tokens are known to Vader, which interestingly is greater than the 13% for the general dataset. However, by performing a Mann-Whitney U test we see that there is no statistically significant difference between the ranks of the known and unknown tokens, meaning that the known tokens are evenly distributed. Therefore, it is reasonable to believe that the performance difference is likely due to these models finding poor vectors for less common tokens. For both SGNS and 2T-SGNS, the 5^{th} neighbour is selected more than the 1^{st} neighbour, which is an unexpected result, the closer tokens are in the vector space, the stronger the expected semantic similarity. Finally, the trend where the 5T-SGNS model outperforms SGNS for lower-ranked tokens may continue outside of the top 100 tokens.

Analysing the Clusters

It is not initially clear why the models have two clusters. This property is remarkable, given that prosaic text word vectors [2] form a single cluster. While all models can be split into two clusters, each has differently sized and shaped clusters, e.g. 5T-SGNS has a larger disparity between the two clusters than the other models. It may be that relaxing the label constraint allows for a less fragmented vector space because some token pairs may never appear in the same message, despite them appearing temporally close together. The crescent shape cluster is likely an artifact of the UMAP process rather than a feature of the vector space.

Figure 6 shows that the biggest difference between the clusters is how popular a token is, both in terms of 'Token Rank', (overall popularity) and 'Document Rank' (how many documents each token appears in). Cluster 0 tokens tend to be more important to the documents they appear in, despite being less used. It appears that message length is reasonably consistent between the two clusters but cluster 0 tokens appear more often when chat is slower. Overall, we see that cluster 0 tokens are less popular in general but often have a high tf-idf and a flatter distribution across median stream rank, which we interpret to mean that cluster 0 tokens are probably tokens which are specific to certain streamers or games, possibly indicating game-specific terms or personalised emotes. Cluster 1, on the other hand, appears to made up of more platform-wide terms, as they are more popular terms that appear in more documents.

Conclusions and Future Work

This work presents a large-scale livestream chat dataset alongside a case study. The learned vector spaces are shaped in strange ways and small changes to the model, e.g. varying the 'c' value in temporal models, results in very different spaces. Furthermore, these spaces are shaped differently to vector spaces generated from prosaic text, even those trained with SGNS. Clustering these models shows that in general two clusters are found and that the differentiating factor for these clusters seems to be how the tokens are used, through measures such as token rank, document rank and tf-idf.

There is a multitude of potential future work and the authors hope that this study, alongside the dataset, sparks conversation and interest into token vector models for livestream chat. Several key challenges exist, for example, further understanding these vector spaces and research into models which can generate vector spaces with strong semantic or sentimental links between tokens, potentially uncovering the meaning of livestream specific tokens and emotes. Likewise, given these spaces are clusterable, it may be possible to explore the homogeneity of these clusters, e.g. through Hopkins Statistic (Banerjee and Dave 2004), as well as track the popularity of tokens from each cluster over time and from that uncover information about what is happening in the stream and how the audience is reacting. Another avenue of research is to explore the implication of context on token use, especially given that distinct communities form around channels (Hamilton, Garretson, and Kerne 2014; Seering, Kraut, and Dabbish 2017). Finally, suitable word vectorization models can be utilized for downstream NLP tasks such as modelling in-steam events via chat reaction.

Acknowledgments

This work is funded by the EPSRC Centre for Doctoral Training in Intelligent Games and Game Intelligence (IGGI), EP/L015846/1 and the Digital Creativity Labs funded by EPSRC/AHRC/Innovate UK, EP/M023265/1.

[2] Observed through models in Tensorflow's Projector. https://projector.tensorflow.org/.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bakarov, A. 2018. A Survey of Word Embeddings Evaluation Methods. *arXiv:1801.09536 [cs]*. arXiv: 1801.09536.

Banerjee, A., and Dave, R. N. 2004. Validating clusters using the hopkins statistic. In *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, volume 1, 149–153 vol.1.

Barbieri, F.; Espinosa-Anke, L.; Ballesteros, M.; Soler-Company, J.; and Saggion, H. 2017a. Towards the Understanding of Gaming Audiences by Modeling Twitch Emotes. *Proceedings of the 3rd Workshop on Noisy User-generated Text* 11–20.

Barbieri, F.; Espinosa Anke, L.; Ballesteros, M.; Soler, J.; and Saggion, H. 2017b. Towards the Understanding of Gaming Audiences by Modeling Twitch Emotes. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 11–20. Copenhagen, Denmark: Association for Computational Linguistics.

Blashki, K., and Nichol, S. 2005. Game geek's goss: linguistic creativity in young males within an online university forum (94/\/\3 933k'5 9055oneone). *Australian journal of emerging technologies and society* 3(2):71–80.

Bulygin, D. 2018. Chats of esports broadcasts of Dota 2 : topic modeling approach.

Cheung, G., and Huang, J. 2011. Starcraft from the stands: Understanding the game spectator. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, 763–772. New York, NY, USA: ACM.

Chollet, F., et al. 2015. Keras.

Ford, C.; Gardner, D.; and Liu, C. 2017. Chat Speed OP : Practices of Coherence in Massive Twitch Chat. 858–871.

Hamilton, W. A.; Garretson, O.; and Kerne, A. 2014. Streaming on Twitch : Fostering Participatory Communities of Play within Live Mixed Media.

Hutto, C., and Gilbert, E. 2015. Vader: A parsimonious rule-based model for sentiment analysis of social media text.

Jiang, R.; Qu, C.; Wang, J.; Wang, C.; and Zheng, Y. 2020. Towards extracting highlights from recorded live videos: An implicit crowdsourcing approach. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 1810–1813.

Kaytoue, M.; Silva, A.; and Cerf, L. 2012. Watch me playing, i am a professional: a first study on video game live streaming. *Proceedings of the 21st international conference companion on World Wide Web* 1181–1188.

Ljubešić, N., and Fišer, D. 2016. A Global Analysis of Emoji Usage. In *Proceedings of the 10th Web as Corpus Workshop*, 82–89. Berlin: Association for Computational Linguistics.

Loper, E., and Bird, S. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, 63–70. Stroudsburg, PA, USA: Association for Computational Linguistics.

McInnes, L.; Healy, J.; Saul, N.; and Großberger, L. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* 3(29):861.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*. arXiv: 1301.3781.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. 3111–3119.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Mimno, D., and Thompson, L. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2873–2878. Copenhagen, Denmark: Association for Computational Linguistics.

Musabirov, I.; Bulygin, D.; and Okopny, P. 2017. Between an Arena and a Sports Bar : Online Chats of Esports Spectators.

Nakandala, S.; Ciampaglia, G.; Su, N.; and Ahn, Y.-Y. 2017. Gendered conversation in a social game-streaming platform.

Nascimento, G., and Ribeiro, M. 2014. Modeling and Analyzing the Video Game Live-Streaming Community. 1–9.

Olejniczak, J. 2015. A Linguistic Study Of Language Variety Used On Twitch . Tv : Desriptive And Corpus-based Approaches. 2012(May):21–23.

Park, J.; Barash, V.; Fink, C.; and Cha, M. 2013. Emoticon style: Interpreting differences in emoticons across cultures.

Poyane, R. 2018. Toxic communication during streams on twitch.tv. the case of dota 2. In *Proceedings of the 22nd International Academic Mindtrek Conference*, Mindtrek '18, 262–265. New York, NY, USA: Association for Computing Machinery.

Recktenwald, D. 2017. ScienceDirect Toward a transcription and analysis of live streaming on Twitch. *Journal of Pragmatics* 115:68–81.

Schnabel, T.; Labutov, I.; Mimno, D.; and Joachims, T. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 298–307. Lisbon, Portugal: Association for Computational Linguistics.

Seering, J.; Kraut, R. E.; and Dabbish, L. 2017. Shaping Pro and Anti-Social Behavior on Twitch Through Moderation and Example-Setting. 111–125.

Smith, T.; Obrist, M.; and Wright, P. 2013. Live-streaming changes the (video) game. *Proceedings of the 11th european conference on Interactive TV and video - EuroITV '13* 131.

Turney, P. D., and Pantel, P. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research* 37:141–188. arXiv: 1003.1141.

Wiseman, S., and Gould, S. J. J. 2018. Repurposing Emoji for Personalised Communication: Why means "I love you". In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 1–10. Montreal QC, Canada: ACM Press.

Zipf, G. K. 1935. *The psycho-biology of language*. Oxford, England: Houghton, Mifflin. Pages: ix, 336.