Character-based Neural Embeddings for Tweet Clustering

Svitlana Vakulenko
Vienna University of Economics and Business, MODUL Technology GmbH
svitlana.vakulenko@wu.ac.at

Lyndon Nixon
MODUL Technology GmbH
nixon@modultech.eu

Mihai Lupu
TU Wien
Vienna, Austria
mihai.lupu@tuwien.ac.at

Abstract

In this paper we show how the performance of tweet clustering can be improved by leveraging character-based neural networks. The proposed approach overcomes the limitations related to the vocabulary explosion in the word-based models and allows for the seamless processing of the multilingual content. Our evaluation results and code are available on-line[1].

1 Introduction

Our use case scenario, as part of the InVID project[2], originates from the needs of professional journalists responsible for reporting breaking news in a timely manner. News often appear on social media exclusively or right before they appear in the traditional news media. Social media is also responsible for the rapid propagation of inaccurate or incomplete information (rumors). Therefore, it is important to provide efficient tools to enable journalists rapidly detect breaking news in social media streams (Petrovic et al., 2013).

The SNOW 2014 Data Challenge provided the task of extracting newsworthy topics from Twitter. The results of the challenge confirmed that the task is ambitious: The best result was 0.4 F-measure.

Breaking-news detection involves 3 subtasks: selection, clustering, and ranking of tweets. In this paper, we address the task of tweet clustering as one of the pivotal subtasks required to enable effective breaking news detection from Twitter.

Traditional approaches to clustering textual documents involve construction of a document-term matrix, which represents each document as a bag-of-words. These approaches also require language-specific sentence and word tokenization.

Word-based approaches fall short when applied to social media data, e.g., Twitter, where a lot of infrequent or misspelled words occur within very short documents. Hence, the document representation matrix becomes increasingly sparse.

One way to overcome sparseness in a tweet-term matrix is to consider only the terms that appear frequently across the collection and drop all the infrequent terms. This procedure effectively removes a considerable amount of information content. As a result, all tweets that do not contain any of the frequent terms receive a null-vector representation. These tweets are further ignored by the model and cannot influence clustering outcomes in the subsequent time intervals, where the frequency distribution may change, which hinders the detection of emerging topics.

Artificial neural networks (ANNs) allow to generate dense vector representation (embeddings), which can be efficiently generated on the word- as well as character levels (dos Santos and Zadrozny, 2014; Zhang et al., 2015; Dhingra et al., 2016). The main advantage of the character-based approaches is their language-independence, since they do not require any language-specific parsing.

The major contribution of our work is the evaluation of the character-based neural embeddings on the tweet clustering task. We show how to employ character-based tweet embeddings for the task of tweet clustering and demonstrate in the experimental evaluation that the proposed approach significantly outperforms the current state-of-the-art in tweet clustering for breaking news detection.

The remaining of this paper is structured as follows: Section 2 provides an overview of the related work; we describe the setup of an extensive evaluation in Section 3; report and discuss the results in Sections 4 and 5, respectively; conclu-

[1] https://github.com/vendi12/tweet2vec_clustering

[2] http://www.invid-project.eu

Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media , pages 36–44,
Valencia, Spain, April 3-7, 2017.

sion (Section 6) summarizes our findings and directions for future work.

2 Related Work

2.1 Breaking news detection

There has been a continuous effort over the recent years to design effective and efficient algorithms capable of detecting newsworthy topics in the Twitter stream (Hayashi et al., 2015; Ifrim et al., 2014; Vosecky et al., 2013; Wurzer et al., 2015). These current state-of-the-art approaches build upon the bag-of-words document model, which results in high-dimensional, sparse representations that do not scale well and are not aware of semantic similarities, such as paraphrases.

The problem becomes evident in case of tweets that contain short texts with a long tail of infrequent slang and misspelled words. The performance of the such approaches over Twitter datasets is very low, with F-measure up to 0.2 against the annotated Wikipidea articles as reference topics (Wurzer et al., 2015) and 0.4 against the curated topic pool (Papadopoulos et al., 2014).

2.2 Neural embeddings

Artificial neural networks (ANNs) allow to generate dense vector representations (embeddings). Word2vec (Mikolov et al., 2013) is by far the most popular approach. It accumulates the co-occurrence statistics of words that efficiently summarizes their semantics.

Brigadir et al. (2014) demonstrated encouraging results using the word2vec Skip-gram model to generate event timelines from tweets. Moran et al. (2016) achieved an improvement over the state-of-the-art first story detection (FSD) results by expanding the tweets with their semantically related terms using word2vec.

Neural embeddings can be efficiently generated on the character level as well. They repeatedly outperformed the word-level baselines on the tasks of language modeling (Kim et al., 2016), part-of-speech tagging (dos Santos and Zadrozny, 2014), and text classification (Zhang et al., 2015). The main advantage of the character-based approach is its language-independence, since it does not depend on any language-specific preprocessing.

Dhingra et al. (2016) proposed training a recurrent neural network on the task of hashtag prediction. Vosoughi et al. (2016) demonstrated an improved performance of a character-based neural autoencoder on the task of paraphrase and semantic similarity detection in tweets.

Our work extends the evaluation of the Tweet2Vec model (Dhingra et al., 2016) to the tweet clustering task, versus the traditional document-term matrix representation. To the best of our knowledge, this work is the first attempt to evaluate the performance of character-based neural embeddings on the tweet clustering task.

3 Experimental Evaluation

3.1 Dataset

Description and preprocessing. We use the SNOW 2014 test dataset (Papadopoulos et al., 2014) in our evaluation. It contains the IDs of about 1 million tweets produced within 24 hours.

We retrieved 845,626 tweets from the Twitter API, since other tweets had already been deleted from the platform. The preprocessing procedure: remove RT prefixes, urls and user mentions, bring all characters to lower case and separate punctuation with spaces (the later is necessary only for the word-level baseline).

The dataset is further separated into 5 subsets corresponding to the 1-hour time intervals (18:00, 22:00, 23:15, 01:00 and 01:30) that are annotated with the list of breaking news topics. In total, we have 48,399 tweets for clustering evaluation; the majority of them (42,758 tweets) are in English.

The dataset comes with the list of the breaking news topics. These topics were manually selected by the independent evaluators from the topic pool collected from all challenge participants (external topics). The list of topics contains 70 breaking news headlines extracted from tweets (e.g., "The new, full Godzilla trailer has roared online"). Each topic is annotated with a few (at most 4) tweet IDs, which is not sufficient for an adequate evaluation of a tweet clustering algorithm.

Dataset extension. We enrich the topic annotations by collecting larger tweet clusters using fuzzy string matching[3] for each of the topic labels. Fuzzy string matching uses the Levenstein (edit) distance (Levenshtein, 1966) between the two input strings as the measure of similarity. Levenstein distance corresponds to the minimum number of character edits (insertions, deletions, or substitutions) required to transform one string into the

[3] https://github.com/seatgeek/fuzzywuzzy

other. We choose only the tweets for which the similarity ratio with the topic string is greater than 0.9 threshold.

A sample tweet cluster produced with the fuzzy string matching for the topic "Justin Trudeau apologizes for Ukraine joke":

- Justin Trudeau apologizes for *Ukraine joke*: Justin Trudeau said he's spoken the head...
- Justin Trudeau apologizes for *Ukraine comments* `http://t.co/7ImWTRONXt`
- Justin Trudeau apologizes for *Ukraine hockey joke* #cdnpoli

In total, we matched 2,585 tweets to 132 clusters using this approach. The resulting tweet clusters represent the ground-truth topics within different time intervals. The cluster size varies from 1 to 361 tweets with an average of 20 tweets per cluster (median: 6.5).

This simple procedure allows us to automatically generate high-quality partial labeling. We further use this topic assignment as the ground-truth class labels to automatically evaluate different flat clustering partitions.

3.2 Tweet representation approaches

TweetTerm. Our baseline is the tweet representation approach that was used in the winner-system of SNOW 2014 Data Challenge[4] (Ifrim et al., 2014). This approach represents a collection of tweets as a tweet-term matrix by keeping the bigrams and trigrams that occur at least in 10 tweets.

Tweet2Vec. This approach includes two stages: (1) *training* a neural network to predict hashtags using the subset of tweets that contain hashtags (88,148 tweets in our case); (2) *encoding*: use the trained model to produce tweet embeddings for all the tweets regardless whether they contain hashtags or not. We use Tweet2Vec implementation[5] to produce tweet embeddings.

Tweet2Vec is a bi-directional recurrent neural network that consumes textual input as a sequence of characters. The network architecture includes two Gated Recurrent Units (GRUs) (Cho et al., 2014): forward and backward GRUs. GRU is an optimized version of a Long Short-Term Memory (LSTM) architecture (Hochreiter and Schmidhuber, 1997). It includes 2 gates that control the information flow. The gates (reset and update gate) regulate how much the previous output state (h_{t-1}) influences the current state (h_t).

The two GRUs are identical, but the backward GRU receives the same sequence of tweet-characters in reverse order. Each GRU computes its own vector-representation for every substring (h_t) using the current character vector (x_t) and the vector-representation it computed a step before (h_{t-1}). These two representations of the same tweet are combined in the next layer of the neural network to produce the final tweet embedding (see more details in Dhingra et.al. (2016)).

The network is trained in minibatches with an objective function to predict the previously removed hashtags. A hashtag can be considered as the ground-truth cluster label for tweets. Therefore, the network is trained to optimize for the correct tweet classification, which corresponds to a supervised version of the tweet clustering task annotated with the cluster assignment, i.e. hashtags.

In order to predict the hashtags the tweet embeddings are passed through the linear layer, which produces the output in the size of the number of hashtags, which we observed in the training dataset. The softmax layer on top normalizes the scores from the linear layer to generate the hashtag probabilities for every input tweet.

Tweet embeddings are produced by passing the tweets through the trained Tweet2Vec model (encoder). In this way we can obtain vector representations for all the tweets including the ones that do not contain any hashtags. The result is a matrix of size $n \times h$, where n is the number of tweets and h is the number of hidden states (500).

3.3 Clustering

To cluster tweet vectors (character-based tweet embeddings produced by the neural network for Tweet2Vec evaluation or the document-term matrix for TweetTerm) we employ the hierarchical clustering algorithm implementation from *fastcluster* library (Müllner, 2013).

Hierarchical clustering includes computing pairwise distances between the tweet vectors, followed by their linkage into a single dendrogram. There are several distance metrics (Euclidean, Manhattan, cosine, etc.) and linkage methods to compare distances (single, average, complete, weighted, etc.). We evaluated the performance of different methods using the cophenetic correlation

[4] `https://github.com/heerme/twitter-topics`

[5] `https://github.com/bdhingra/tweet2vec`

coefficient (CPCC) (Sokal and Rohlf, 1962) and found the best performing combination: Euclidean distance and average linkage method.

The hierarchical clustering dendrogram can produce n different flat clusterings for the same dataset: from *n* single-member clusters with one document per cluster to a single cluster that contains all *n* documents. The distance threshold defines the granularity (number and size) of the produced clusters.

3.4 Distance threshold selection

Grid search helps us to determine the optimal distance threshold for the dendrogram cut-off. We generated a list of values in the range from 0.1 to 1.5 with 0.1 increment step and examine their performance with respect to the ground-truth cluster assignment. We produce flat clusterings for each value of the distance threshold from the grid and compare them with respect to the quality metrics.

Since we also want to be able to select the optimal distance threshold in absence of the true labels, we examine the scores provided by the mean Silhouette coefficient (Rousseeuw, 1987). Silhouette is an unsupervised intrinsic evaluation metric (cluster validity index) that measures the quality of the produced clusters and can be used for unsupervised intrinsic evaluation (i.e., without the ground-truth labels). It was reported to outperform alternative methods in a comparative study of 30 validity indices (Arbelaitz et al., 2013).

3.5 Clustering Evaluation Metrics

We evaluate the clustering results using the standard metrics for extrinsic clustering evaluation: homogeneity, completeness, V-Measure (Rosenberg and Hirschberg, 2007), Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and Adjusted Mutual Information (AMI) (Nguyen et al., 2010). All metrics return a score on the range [0; 1] for the pair of sets that contain ground truth and cluster labels as input. The higher the score the more similar the two clusterings are.

The **Homogeneity** score represents the measure for purity of the produced clusters. It penalizes clustering, where members of different classes get clustered together. Thus, the best homogeneity scores are always at the bottom of the dendrogram, i.e., at the level of the leaves, where each document belongs to its own cluster. **Completeness**, on the contrary, favors larger clusters and reduces the score if the members of the same class are split into different clusters. Therefore, the top of the dendrogram, where all the documents reside in a single cluster always achieves the maximum completeness score.

V-Measure is designed to balance out the two extremes of homogeneity and completeness. It is the harmonic mean of the two and corresponds to the Normalized Mutual Information (NMI) score.

AMI score is an extension of NMI adjusted for chance. The more clusters are considered the more chance the labelings correlate. AMI allows us to compare the clustering performance across different time intervals since it normalizes the score by the number of labeled clusters in each interval.

Finally, **ARI** is an alternative way to assess the agreement between two clusterings. It counts all pairs clustered together or separated in different clusters. ARI also accounts for the chance of an overlap in a random label assignment.

3.6 Manual Cluster Evaluation

Our partial labeling covers a small subset of the data and by design provides the clusters with the high degree of string overlap with the annotated topics. Therefore, we extend the clustering evaluation to the rest of the dataset to evaluate whether the models can uncover less straight-forward semantic similarities in tweets. We select the results for manual evaluation motivated by the cluster label (headline) selection task.

The next step in the breaking news detection pipeline after the clustering task is headline selection (cluster labeling task). The most common approach to label a cluster of tweets is to select a single tweet as a representative member for the whole cluster (Papadopoulos et al., 2014). We decided to test this assumption and manually check how many clusters loose their semantics when represented with a single tweet.

Headline selection motivates the coherence assessment of the produced clusters since the clusters discarded at this stage will never make it to the final results. To explore coherence of the produced clusters we pick several tweets in each cluster and check whether they are semantically similar.

The tweet selected as a headline (cluster label) can be the first published tweet as in First Story Detection (FSD) task, also used in Ifrim et al (2014). Alternative approaches include selection of the most recent tweet published on the topic, or the tweet that is semantically most similar

Interval	Tweets	Model	Dimensions	Distance threshold	Clusters	Homogeneity	Completeness	V-Measure	ARI	AMI
18:00	10,344	Tweet2Vec	500	1	3026	**0.9958**	0.9453	**0.9699**	**0.9804**	**0.9376**
		TweetTerm	433	1-1.3	66-79	0.9277	**1**	0.9625	0.949	0.9216
22:00	14,471	Tweet2Vec	500	0.9	5292	**1**	0.9601	**0.9796**	**0.9922**	**0.9571**
		TweetTerm	589	0.7-1.3	93-118	0.9385	**0.9969**	0.9668	0.9859	0.9359
23:15	8,231	Tweet2Vec	500	0.8	3986	**1**	0.98	**0.9899**	**0.9948**	**0.9743**
		TweetTerm	565	0.01-1.3	67-142	0.8062	**0.9978**	0.8918	0.7344	0.7763
01:00	5,123	Tweet2Vec	500	0.9	2242	**1**	0.8877	**0.9405**	**0.8668**	**0.8327**
		TweetTerm	721	0.8-1.3	71-111	0.8104	**1**	0.8953	0.8188	0.7666
01:30	4,589	Tweet2Vec	500	0.9	2091	**1**	0.8762	**0.934**	**0.8089**	**0.8129**
		TweetTerm	635	1.2-1.3	64-78	0.8024	**1**	0.8903	0.7809	0.754

Table 1: Results of clustering evaluation on the English-language dataset

to all other tweets in the cluster, i.e., the tweet closest to the centroid of the cluster (**medoid-tweets**). Therefore, we sample 5 tweets from each cluster: the first published tweet, the most recent tweet and three medoid-tweets.

We set up a manual evaluation task as follows:

1. Take the top 20 largest clusters sorted by the number of tweets that belong to the cluster.
2. For each cluster:
   (a) Take the first and the last published tweet (tweets are previously sorted by the publication date).
   (b) Take three medoid-tweets, i.e., the tweets that appear closest to the centroid of the cluster.
   (c) Add the 5 tweets to the set associated with the cluster (removing exact duplicate tweets)
3. For all clusters, where the set of selected tweets contains at least two unique tweets: 4 human evaluators independently assess the coherence of each cluster.

According to the evaluation setup each model produced 20 top-clusters for each of the 5 intervals, i.e., $20 \times 5 = 100$ clusters per model. We manually evaluate only the clusters that contain more than 1 distinct representative tweet (**Clusters>1**). All other clusters, i.e., the ones for which all 5 selected tweets are identical (**Clusters=1**), are considered correct by default.

Results for all 5 intervals were evaluated together in a single pool and the models were anonymized to avoid biases. Each evaluator independently assigned a single score to each cluster:

- **Correct** – all tweets report the same news;
- **Partial** – some tweets are not related;
- **Incorrect** – all tweets are not related.

Partial and Incorrect labels reflect different types of clustering errors. Partial error is less severe indicating that the tweets of the cluster are semantically similar, but they report different news (events) and should be split into several clusters. Incorrect clusters indicate a random collection of tweets with no semantic similarities.

4 Results

4.1 Results of Clustering Evaluation

Table 1 summarizes the results of our evaluation using the ground-truth partial labeling. The scores highlighted with the bold font indicate the best result among the two competing approaches for the same subset of tweets corresponding to the respective time interval.

Tweet2Vec exhibits better clustering performance comparing to the baseline according to the majority of the evaluation metrics in all the intervals. In all cases Tweet2Vec model wins in terms of Homogeneity score and TweetTerm wins in Completeness. This result shows that Tweet2Vec is better at separating tweets that are not similar enough than the baseline model. Tweet2Vec fails only once to perfectly separate the ground-truth clusters (18:00 interval). This result shows that Tweet2Vec is able to replicate the results of the fuzzy string matching algorithm that was used to generate the ground-truth labeling.

4.2 Results of Distance Threshold Selection

The rise in V-Measure correlates with the decline of the Silhouette coefficient and the steep drop in the number of produced clusters (see Figure 1). We observed that the optimal distance threshold for Tweet2Vec clustering according to V-Measure is on the interval [0.8; 1] (see Table 1: Distance threshold), which is also consistent with the findings reported in Ifrim et. al (2014).

Model	Dataset	Clusters	Correct (%)			Errors (%)	
			Clusters=1	Clusters>1	Total	Partial	Incorrect
Tweet2Vec	English	100	**80**	8.3	**88.3**	10	**1.8**
TweetTerm	English	95	71	**17.4**	87.9	8.9	3.2
Tweet2Vec	Multilingual	100	67	12.5	**79.5**	13	7.5

Table 2: Results of manual cluster evaluation. Note: the last row shows results on a different dataset and can not be directly compared with the other models.

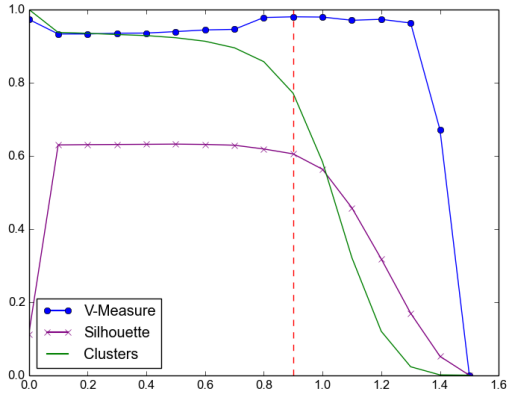

Figure 1: Correlation between the V-Measure, Silhouette coefficient and the number of clusters per tweet (Tweet2Vec 22:00 interval). The vertical red line indicates the maximum V-Measure score.

4.3 Results of Manual Cluster Evaluation

Results of the manual cluster evaluation by four independent evaluators are summarized in Table 2. Bold font indicates the maximum scores achieved across the competing representation approaches. Tables 3 and 4 show sample clusters produced by both models alongside their average score.

TweetTerm assigns a 0-vector representation to tweets that do not contain any of the frequent terms. Hence, all these tweets end up in a single "garbage" cluster. Therefore, we discount the number of the expected "garbage" clusters (1 cluster per interval = 5 clusters) from the score count for TweetTerm (Table 2).

Tweet2Vec model produces the largest number of perfectly homogeneous clusters for which all 5 selected tweets are identical (see Table 2 column Clusters=1). The percentage of correct results among the manually evaluated clusters is higher for the TweetTerm model, but the number of errors (Incorrect) is higher as well. Tweet2Vec produced the highest total % of correct clusters due to the larger proportion of detected clusters that contain identical tweets (Clusters=1). Tweet2Vec also produced the least number of incorrect clusters: at most 2 incorrect clusters per 100 clusters (Precision: 0.98).

The results of Tweet2Vec on the multilingual dataset are lower than on the English-language tweets. However, we do not have alternative results to compare since the baseline approach is not language-independent and requires additional functionality (word-level tokenizers) to handle tweets in other languages, e.g., Arabic or Chinese. We provide this evaluation results to demonstrate that Tweet2Vec overcomes this limitation and is able to cluster tweets in different languages. In particular, we obtained correct clusters of Russian and Arabic tweets.

We observed that leaving the urls does not significantly affect clustering performance, i.e., the model tolerates noise. However, replacement of the urls and user mentions with placeholders as in Dhingra et. al. (2016) generates syntactic patterns in text, such as @user @user @user, which causes semantically unrelated tweets appear within the same cluster.

5 Discussion

Our experimental evaluation showed that the character-based embeddings produced with a neural network outperform the document-term baseline on the tweet clustering task. The baseline approach (TweetTerm) shows a very good performance in comparison with the simplicity of its implementation, but it naturally falls short in recognizing patterns beyond simple n-gram matching.

We attribute this result to the inherent limitation of the document-term model retaining only the frequent terms and disregarding the long tail of infrequent patterns. This limitation appears crucial in the task of emergent news detection, in which the topics need to be detected long before they become popular. Neural embeddings, in contrast, can retain a sufficient level of detail in their representa-

Sample Cluster	Evaluation
video : **bitcoin : mtgox** exchange goes offline - **bitcoin** , a virtual currency ... the slow-motion collapse of **mt . gox** is **bitcoin**'s first financial crisis **:** now **bitcoin** users ... Disastro **bitcoin** : **mt . gox** cessa ogni attivite ... **: mt . gox** , il pi grande cambiavalute **bitco** ...	Correct
california couple finds time capsules **worth $10 million** **californian couple finds $10 million worth** of gold coins in tin can	Correct
ukraine puts off vote on new government despite eu pleas for quick action **-** washington post ... **ukraine** truce shattered , death toll hits 67 **-** kiev (reuters) **-** ukraine suffered its bloodiest day ... **ukraine** fighting leaves at least 18 dead as kiev barricades burn **-** clashes in ukraine ...	Partial
are you going to come on his network and get poor ratings too **?** **are you** sold on the waffle taco **?**	Incorrect
the chromecast app flood has started **by** **the** importance of emotion in design **by**	Incorrect

Table 3: Tweet2Vec sample results. Rows of the table show sample tweet clusters. Each line within the row corresponds to a separate tweet (after preprocessing, i.e. usernames and urls removed.)

Sample Cluster	Evaluation
obama : michelle and i were saddened to hear of the passing of **harold ramis**... touching tribute to ghostbusters star **harold ramis** from comic artist on the joyful comedy of **harold ramis**	Correct
major tokyo-based bitcoin exchange **mt . gox goes dark** "bitcoin exchange giant **mt . gox goes dark** — popular science "	Correct
obesity rate for young children plummets **43 %** in a **decade** the national **obesity rate for young children** dropped **43 %** over the past **decade**	Correct
diplomatic pressure is unlikely to reverse uganda's cruel **anti-gay** law provisions of arizona proposed **anti-gay** law even mitt romney wants arizona's governor to veto the state's **anti-gay** bill icymi : arizona pizzeria response to state **anti-gay** bill	Partial
amazing debate nic ! **well done** ! **well done** 4 -0 **well done** ! i find running so difficult . feel proud ! **well done** him :-) **well done** nicola my money is on you you done it well tonight ??	Incorrect

Table 4: TweetTerm sample results. Rows of the table show sample tweet clusters.

tions and are able to mirror the fuzzy string matching performance beyond simple n-gram matching.

It becomes apparent from the sample clustering results (Tables 3 and 4) that both models perform essentially the same task of unveiling patterns shared between a group of strings. While TweetTerm operates only on the patterns of identical n-grams, Tweet2Vec goes beyond this limitation by providing room for a variation within the n-gram substring similar to fuzzy string matching. This effect allows to capture subtle variations in strings, e.g., misspellings, which word-based approaches are incapable of.

Our error analysis also revealed the limitation of the neural embeddings to distinguish between semantic and syntactic similarity in strings (see Incorrect samples in Table 3). Tweet2Vec, as a recurrent neural network approach, represents not only the characters but also their order in string that may be a false similarity signal. It is evident that the neural representations in our example would benefit from the stop-word removal or an analogous to TF/IDF weighting scheme to avoid capturing punctuation and other merely syntactic patterns.

Limitations. Neural networks gain performance when more data is available. We could use only 88,148 tweets from the dataset to train the neural network, which can appear insufficient to unfold the potential of the model to recognize more complex patterns. Also, due to the scarce annotation available we could use only a small subset of the original dataset for our clustering evaluation. Since most of the SNOW tweets are in English, another dataset is needed for comprehensive multilingual clustering evaluation.

6 Conclusion

We showed that character-based neural embeddings enable accurate tweet clustering with minimum supervision. They provide fine-grained representations that can help to uncover fuzzy similarities in strings beyond simple n-gram matching. We also demonstrated the limitation of the current

approach unable to distinguish semantic from syntactic patterns in strings, which provides a clear direction for the future work.

7 Acknowledgments

The presented work was supported by the InVID Project (http://www.invid-project.eu/), funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 687786. Mihai Lupu was supported by Self-Optimizer (FFG 852624) in the EUROSTARS programme, funded by EUREKA, the BMWFW and the European Union, and ADMIRE (P25905-N23) by FWF. We thank to Bhuwan Dhingra for the support in using Tweet2Vec and Linda Andersson for the review and helpful comments.

References

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona. 2013. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256.

Igor Brigadir, Derek Greene, and Padraig Cunningham. 2014. Adaptive Representations for Tracking Breaking News on Twitter. In *NewsKDD - Workshop on Data Science for News Publishing at The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, August 24-27, 2014, New York, NY, USA*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*, pages 1724–1734.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W. Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany*.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, 21-26 June, 2014, Beijing, China*, pages 1818–1826.

Kohei Hayashi, Takanori Maehara, Masashi Toyoda, and Ken-ichi Kawarabayashi. 2015. Real-Time Top-R Topic Detection on Twitter with Topic Hijack Filtering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 10-13, 2015, Sydney, Australia*, pages 417–426.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Georgiana Ifrim, Bichen Shi, and Igor Brigadir. 2014. Event Detection in Twitter using Aggressive Filtering and Hierarchical Tweet Clustering. In Symeon Papadopoulos, David Corney, and Luca Maria Aiello, editors, *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), April 8, 2014, Seoul, Korea*, pages 33–40.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2741–2749.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In Christopher J. C. Burges, Lon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Sean Moran, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2016. Enhancing First Story Detection using Word Embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, July 17-21, 2016, Pisa, Italy*, pages 821–824.

Daniel Müllner. 2013. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(1):1–18.

Xuan Vinh Nguyen, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854.

Symeon Papadopoulos, David Corney, and Luca Maria Aiello. 2014. SNOW 2014 Data Challenge: Assessing the Performance of News Topic Detection Methods in Social Media. In Symeon Papadopoulos, David Corney, and Luca Maria Aiello, editors,

Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), April 8, 2014, Seoul, Korea, pages 1–8.

Sasa Petrovic, Miles Osborne, Richard McCreadie, Craig Macdonald, Iadh Ounis, and Luke Shrimpton. 2013. Can twitter replace newswire for breaking news? In Emre Kiciman, Nicole B. Ellison, Bernie Hogan, Paul Resnick, and Ian Soboroff, editors, *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, July 8-11, 2013, Cambridge, Massachusetts, USA*.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 410–420.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65.

Robert R. Sokal and F. James Rohlf. 1962. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40.

Jan Vosecky, Di Jiang, Kenneth Wai-Ting Leung, and Wilfred Ng. 2013. Dynamic multi-faceted topic discovery in twitter. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, October 27 - November 1, 2013, San Francisco, CA, USA*, pages 879–884.

Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. 2016. Tweet2vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, July 17-21, 2016, Pisa, Italy*, pages 1041–1044.

Dominik Wurzer, Victor Lavrenko, and Miles Osborne. 2015. Tracking unbounded Topic Streams. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China*, pages 1765–1773.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.