Proceedings of the 12th Web as Corpus Workshop, pages 57–64
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Streaming Language-Specific Twitter Data with Optimal Keywords

Tim, Kreutz, Walter, Daelemans
CLiPS, University of Antwerp
Prinsstraat 13, 2000 Antwerp, Belgium
tim.kreutz, walter.daelemans@uantwerpen.be

Abstract

The Twitter Streaming API has been used to create language-specific corpora with varying degrees of success. Selecting a filter of frequent yet distinct keywords for German resulted in a near-complete collection of German tweets. This method is promising as it keeps within Twitter endpoint limitations and could be applied to other languages besides German. But so far no research has compared methods for selecting optimal keywords for this task. This paper proposes a method for finding optimal key phrases based on a greedy solution to the maximum coverage problem. We generate candidate key phrases for the 50 most frequent languages on Twitter. Candidates are then iteratively selected based on a variety of scoring functions applied to their coverage of target tweets. Selecting candidates based on the scoring function that exponentiates the precision of a key phrase and weighs it by recall achieved the best results overall. Some target languages yield lower results than what could be expected from their prevalence on Twitter. Upon analyzing the errors, we find that these are languages that are very close to more prevalent languages. In these cases, key phrases that limit finding the competitive language are selected, and overall recall on the target language also decreases. We publish the resulting optimized lists for each language as a resource. The code to generate lists for other research objectives is also supplied.

Keywords: twitter, social media, data collection, corpus linguistics

1. Introduction

Twitter data has frequently been used to study the public reaction to specific topics or events (Leetaru et al., 2013). In Natural Language Processing this trend is mirrored in popular subtasks like sentiment mining and event detection, and the appeal of tweets for these purposes is understandable; they comprise abundant, open and mostly unfiltered public feedback (Barnaghi et al., 2016).

But collecting tweets for diverse purposes is no straightforward task. Researchers ultimately make design choices on which keywords, hashtags and users to search, without any gold standard reference to test the resulting data snapshot. Additionally, not all tweets are made available to the search index (Twitter, 2019c). Twitter is free to put any restrictions on their results, whether it is on the maximum number of hits or on how far back search results go.

As an alternative to the retrospective search approach, the Twitter Streaming API (Twitter, 2019b) has been used to collect high-volume language-specific corpora in real-time. By filtering the stream on a list of frequent yet distinct keywords for a specific language, it is possible to achieve high coverage of a reference set. Such lists of keywords have been created for Dutch (Tjong Kim Sang and Van den Bosch, 2013), Italian (Basile and Nissim, 2013), German (Scheffler, 2014), Hindi, Telugu and Bengali (Choudhary et al., 2018).

This paper offers three main improvements to the previous work. First, we compare methods for selecting optimal keywords for creating language-specific Twitter corpora. Second, we closely replicate the real-world performance of these methods in our experimental setup so that the limitations of the resulting corpora are known for any downstream task. Third, although we conform to the Twitter Developer Agreement (Twitter, 2019a) and will not share the language-specific corpora, we do provide the lists of optimized keywords for the top 50 languages on Twitter and the code to generate lists for other languages.

2. Background

Distribution of large collections of tweets is disallowed under the Twitter Developer Agreement and Policy (Twitter, 2019a). Initiatives to share large general-purpose Twitter collection, such as the Edinburgh Twitter Corpus (Petrović et al., 2010) have been shut down under this regulation.

Consequently, studies on Twitter data have moved away from large scale general-purpose collections to data snapshots designed for a specific downstream task. Three main filtering approaches can be distinguished in previous work.

2.1. Location-based Filtering

Twitter introduced an opt-in for sending location information with tweets in 2009. This has allowed researchers to study language use alongside fine-grained geographic distinctions.

Location-based filtering has proven invaluable for creating datasets for dialectology with relatively low effort (Eisenstein et al., 2010; Huang et al., 2016). Laitinen et al. (2018) show that location-based filtering can successfully be deployed for studying language spread across country borders.

Location-based filtering is less suitable for creating language-specific corpora. Bergsma et al. (2012) design filters based on the coordinates of major cities where speakers of a target language are prominent. The resulting collections were relatively pure for Arabic (99.9%) and Farsi (99.7%) but not for Urdu (61.0%). Since a very low percentage (between 0.7% and 2.9% depending on the country) of Twitter users enable location sharing, filtering by location yields very low coverage (Barbaresi, 2016).

2.2. User-based Filtering

Filtering by username is useful in cases where a very specific group of users is targeted. Praet et al. (2018) collected tweets by Flemish politicians to analyze which political issues were most often communicated, and whether

this aligned with their parties' agenda.
Barbaresi (2016) used user-based filtering in conjunction with location-based filtering to find tweets by Austrian Twitter users. The resulting collection had mostly English (42.2%) language tweets.

2.3. Word-based Filtering

Word-based filtering best suits the purpose of creating language-specific corpora. Scheffler (2014) was able to collect a near-complete snapshot of German twitter messages by tapping the Streaming API for German stopwords. They removed words from the list that are also frequent in other languages (such as 'war' and 'die' for English) and extended it with other frequent and distinctive German words. To test for coverage, the collection obtained through word-based filtering was compared to collections retrieved with location-based and user-based filtering during the same time period. Only around 5% of German tweets was missing from the collection obtained through word-based filtering.
Handpicked lists of filter words were also used for collecting Dutch tweets (Tjong Kim Sang and Van den Bosch, 2013). The authors add Twitter-specific terms, such as trending Dutch hashtags, to their keywords but report that a lot of other-language tweets still slip through the filter.
A more systematic approach can be found in Basile and Nissim (2013). Cross-language homographs were detected using Google Ngrams and removed from a list of most frequent Italian words. Using only the top 20 of the remaining terms yielded enough data for the eventual purpose of creating an Italian corpus for sentiment analysis.

2.4. Toward Optimal Filtering

Previous work on word-based filtering has mostly been deployed as an intermediate step for a downstream task. These papers understandably deploy some heuristic method of selecting keywords, and usually do not compare the resulting snapshot with a reference set.
Kreutz and Daelemans (2019) instead focus solely on obtaining optimized keywords. Their list of optimized keywords for Dutch outperforms the hand-picked list in Tjong Kim Sang and Van den Bosch (2013) in both precision and recall. From intrinsic evaluation it is also clear that the optimized list benefits from being generated on the domain it is trying to retrieve.
We extend the work of Kreutz and Daelemans (2019) by comparing additional optimization methods and applying these to languages other than Dutch. In the process of developing optimal lists that can be used to collect language-specific Twitter corpora for the 50 most common languages on Twitter, we provide the statistics that can be cited as limitations for these collections.

3. Data

To generate optimal keywords over Twitter data, we design an experimental setup that mirrors the performance of the keywords on the real-time stream.

3.1. Twitter API Constraints

The Twitter API imposes a 1% rate limit, and will automatically sample down to the rate limit when more tweets pass the filter (Twitter, 2019b). This puts a hard limit on the number of tweets that can be obtained for the more dominant languages on Twitter. Language prevalence can be used to determine the maximum coverage any filtering can achieve.

3.2. Language Prevalence

We collected tweets using the Twitter sprinkler (Twitter, 2019b) over a period of six months from October 2017 to March 2018. The Twitter Sprinkler is an access point of the Twitter Streaming API that can yield 1% of all tweets at any time. Filtering of the complete datastream can be done by giving keyphrases, geo-locations, or user handles. We did not apply any filtering to best approximate a random sample. This resulted in roughly 570 million tweets.
Although Twitter predicts its own IETF language tags for most tweets, we found on initial inspection that a pre-trained FastText language identification model (Joulin et al., 2017) identified a larger part of the tweets. We think it is key to assign labels to difficult and even code mixed tweets. These non-trivial cases crop up in the real-world setting and cannot be ignored for generating keyphrases and for reporting their performance.
The FastText (large) 176 ISO-tag model was used to assign silver labels to each tweet. The tags come from a combination of the ISO 639-1 and ISO 639-2 standards found on the FastText website (Grave, 2017). Table 1 shows the language prevalence of the five most and the five least identified languages. FREQ is the relative frequency over our entire dataset. MEAN is the relative frequency averaged per hour and better reflects language prevalence normalised over time. MEAN can be used to determine how many tweets cannot be retrieved due to the 1% rate limit. MAX shows the maximum hourly relative frequency. Languages that never surpass the 1% rate limit throughout the day can theoretically be collected in full.

Language	FREQ	MEAN	MAX
1. English	39.06%	39.21%	46.93%
2. Japanese	19.18%	19.09%	29.64%
3. Spanish	9.52%	9.45%	13.27%
4. Arabic	7.29%	7.39%	10.82%
5. Portuguese	5.17%	5.10%	9.59%
<40 more languages>			
46. Azerbaijani	0.01%	0.01%	0.02%
47. Marathi	0.01%	0.01%	0.02%
48. Guarani	0.01%	0.01%	0.02%
49. Albanian	0.01%	0.01%	0.01%
50. Kannada	0.01%	0.01%	0.01%

Table 1: Language frequency (FREQ), averaged frequency per hour (MEAN) and maximum average frequency per hour (MAX) for the most and least identified languages in 6 months of Twitter data.

The Table 1 rankings partially correspond to earlier analyses of the language composition of Twitter. Two notable differences are the increase in the number of Arabic tweets, and a decline in English language tweets compared to a

2011 study (Hong et al., 2011). We expect these differences to be due to increased popularity of Twitter in Arabic countries while the U.S. user base stagnated (Semiocast, 2011; Bloomberg, 2019). However, differences can also be due to the FastText model identifying more tweets (roughly 9%) than the IETF labels used in Hong et al. (2011).

3.3. Experimental Setup

After removing retweets, 10,000 tweets were sampled for the 50 most frequent languages. Non-target language tweets were added conform to the language distributions. For example, since Spanish tweets represent roughly 9.52% of the stream, we sampled 105,042 ($\frac{10000}{0.0952}$) other-language tweets. Since more infrequent languages greatly inflate the number of other-language tweets supplemented in their dataset, we opted for a cut-off after the 50 most frequent languages.

We created development and test data in a similar way, by sampling roughly 5,000 target language tweets and adding other language tweets based on their distribution.

While creating separate data sets for each of the targeted languages may seem extraneous, we opted for this approach because it would guarantee that key phrase lists would be sampled from roughly the same number of tweets for each language. This way, the quality of key phrases can never be attributed to differences in data size.

3.4. Preprocessing

In preparation of generating and testing keywords, tweets are parsed according to Twitter documentation (Twitter, 2019a). Tweets were lowercased and any punctuation except @ (for mentions of other users) and # (for hashtag topic markers) were removed.

4. Methods

The Twitter API allows an input of up to 400 60-byte strings. Disjunctive search is performed between the 400 inputs, and any tweet matching the conjunctive presence of tokens in an input is retrieved (Twitter, 2019a). From now on, string inputs will be referred to as *key phrases*.

We generate token *powersets*; exhaustive combinations of tokens present in the target language tweets. The notion is that each token combination generated from a tweet can be used as a key phrase to retrieve that tweet from the stream. Each key phrase is thus associated with a set of target- and other-language tweets, and in extension a recall and precision score.

4.1. Maximum Coverage of Tweets

Optimal key phrases maximally cover the set of target language tweets, whilst limiting the number of other-language tweets retrieved. The latter consideration is especially important considering the 1% rate limit. Key phrases that confuse target language tweets with other (dominant) languages can lead to results that are not only impure, but also incomplete due to down-sampling.

Formally, we consider a collection of key phrases K, generated from a target language l, $K^l = \{K_1^l, K_2^l, ..., K_n^l\}$ and a parallel collection T of sets of tweets identified by those phrases, $T = \{T_1, T_2, ..., T_n\}$. We compare algorithms for selecting up to 400 phrases from K to optimize a variety of objectives to target set T^l.

Input : K; T;
Output: Optimized key phrases O

Function `Optimal(`K, T`)`**:**
  $bestscore \leftarrow 0$
  $bestphrase \leftarrow None$
  for $i \leftarrow 0$ **to** |K| **do**
    if `Score(`T_i`)` $> bestscore$ **then**
      $bestscore \leftarrow$ `Score(`T_i`)`.
      $bestphrase \leftarrow K_i$.
  return $bestphrase$

Function `Run(`K, T, $n \leftarrow 400$`)`**:**
  $O \leftarrow \varnothing$
  for $i \leftarrow 0$ **to** n **do**
    Remove tweets covered by O from every set in T.
    Add `Optimal(`K, T`)` to O.
  return O

Figure 1: Our method iteratively picks a phrase K_i with the highest score with regards to target set T_l and removes all retrieved tweets from the remaining items in T.

4.2. Scoring Functions

In its classic setting a maximum coverage problem optimizes recall over a target set. Since we also care about precision, we design scoring functions to reflect this objective alongside the naive optimization of recall and precision:

1. Optimize Recall (R)
2. Optimize Precision (P)
3. Optimize Recall, but ensure a precision threshold of .9 for each phrase (R_p)
4. Optimize Precision, but ensure a recall threshold of .01 for each phrase (P_r)
5. Weight Precision$^\beta$ by Recall. Higher β adds more importance to precision ($P^\beta * R$)

Although F-score seems like another likely candidate for scoring key phrases, its reliance on a balanced recall and precision, even in adaptations like F-beta where precision receives more weight, make it unsuitable. We demonstrate the pitfall of reliance on recall sufficiently with scoring functions 1 and 4.

4.3. Greedy Selection

We consider only a greedy approach to selecting key phrases, due to the huge number of candidates. Greedy optimization of maximum coverage problems is shown to be the best approximation algorithm in polynomial time (Feige, 1998). The greedy algorithm iteratively picks a key phrase according to a scoring function from the preceding list. The covered tweets are then removed and scores are recalculated before picking the next phrase (Figure 1).

4.4. Baselines

Naive scoring functions 1 and 2 can be expected to perform poorly for the task of creating language-specific Twitter corpora. We expect optimization over recall to select the stopwords that best identify a target language in addition to other generic terms such as partial URLs. Optimizing precision conversely can yield some terms occurring in only a few tweets.

For more reasonable baseline behavior we draw from previous work in word-based filtering of tweets in Section 2.3. First, keyword lists are compiled from the 400 most frequent tokens in a target language training set in line with Choudhary et al. (2018). These lists are then filtered for cross-language homographs for the second baseline. However, making corrections for each language by hand as seen for Dutch (Tjong Kim Sang and Van den Bosch, 2013) and German (Scheffler, 2014) would require significant language expertise and time investment. We instead assure that none of the 400 selected words are present in the 1000 most frequent terms of non-target languages. This automatic filtering of frequent terms is comparable to what has been done for Italian (Basile and Nissim, 2013).

5. Results

In this section we first qualitatively analyze the key phrases selected by the different scoring functions. Some expected drawbacks of each of the greedy selection approaches have been formulated in the previous section, and are tested by manual inspection.

We do not assume that scoring functions perform uniformly for each target language. Specifically, we expect a prevalence effect whereby language that are more common on Twitter would benefit from higher precision phrases as confusion with other languages is more costly. False positives fill up the stream permitted by the Twitter rate limit and would lower overall performance. For rarer languages, this is less important. The $P^{\beta} * R$ scoring function will be grid-searched for individual languages on the development data to choose a β value.

Languages that have drastically different performance from the mean warrant closer inspection with confusion matrices. We hypothesize that languages that have multiple very closely related languages in the data set score lower due to frequent confusion with those languages. Alternatively, relatively bad performance can be due to under-representation in the data. Languages that are less common on Twitter run a higher risk of selecting false positives with their key phrases.

Finally, we compare the best greedy selection algorithm with the proposed baseline methods on the test data.

5.1. Phrase Lists

Consider the outcome for English of the 50 phrases based on recall and precision in Figure 2.

As expected, the top 50 phrases selected based on their recall contain stopwords and partial URLs. We find some other interesting Twitter-specific terms such as the hashtag "*#iheartawards*" and chat speak "*lol*", "*twt*" and "*ng*".

Scored by recall rt, https, co, the, to, you, and, my, is, that, for, it, in, of, me, this, no, on, good, are, lol, so, just, your, #iheartawards, can, na, with, what, not, need, too, happy, hahahaha, hello, at, have, from, new, yes, or, thanks, twt, hahaha, ng, how, bye, up, hi, like

Scored by precision to have, of is rt, the we, rt to on, and that, the from, would, their, of on, the rt it, rt is and, the rt at, https to for, when you, the they, being, to who, the your, for on, the as, into, to are, rt she, is on, my with, should, rt see, of in https, https today, rt than, many, rt get co, to our, https his, rt really, my this, for you co, in just, to was, https these, the an, of to https, rt for and, automatically, the up, does, getting, is not, my rt you, it this

Figure 2: Resulting key phrase lists from optimizing on recall, precision and F-score respectively.

The phrases selected by precision instead contain n-grams that combine stop words with partial URLS and less frequent words that are more distinct for English.

5.2. Prevalence effect

Positioned between the precision and recall scoring functions is the selection procedure that weights precision by itself and by recall. By taking an exponentiation of precision we increase its effect in the optimization function, which may be prudent after seeing the non-distinctive selections by recall in the previous section.

The importance of increasing the weight of precision over recall may differ between languages. Instead of looking at any individual language we test three configurations (P^1, P^2 and P^8) on languages binned by their frequency rank from Table 1.

Figure 3 shows that a $\beta > 1$ increases performance for the most common language on Twitter. In the ranks 20-30, however, scoring key phrases on their precision weighted by recall performs best. There are no big differences between values of β. We opt to use $P^2 * R$ for the top 25 languages and $P * R$ for the less common languages in our final scoring function.

5.3. Development Set Performance

Table 2 lists the macro averaged performance for each of the proposed scoring functions. Besides recall we shows *bound recall*, which is the performance of the key phrases under the Twitter rate limit.

Since optimizing recall yields a lot of non-distinctive terms, the retrieved set of tweets proves impure and recall drops when we take the 1% rate limit into account. This is also the case when optimizing precision but respecting a minimum recall threshold of .01.

The three other scoring functions perform better. Simply selecting key phrases on their precision leads to a high precision overall. The yielded 400 high-precision phrases also cover a reasonably large part of the target language tweets (58.67%). The function that selects phrases on the basis

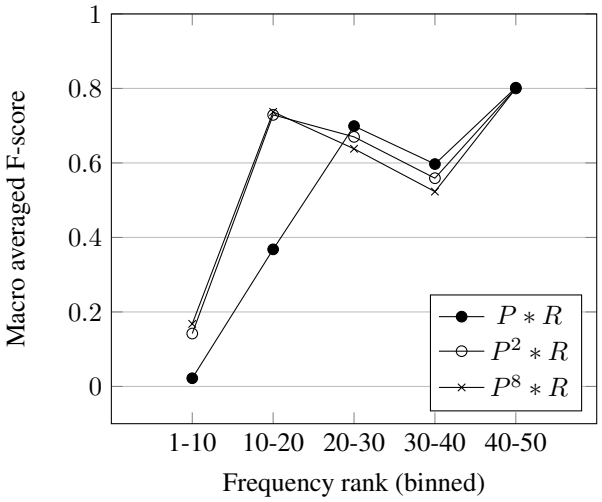

Figure 3: Adding more weight to precision works best for the most prevalent languages on Twitter. Rarer languages benefit from selecting key phrases based on regular precision weighted recall.

Method	Precision	Recall	Bound Recall	F-score
R	16.86%	**85.53**%	1.71%	3.10%
P	**95.17**%	58.76%	45.40%	57.22%
R_p	94.91%	60.85%	45.60%	57.40%
P_r	20.16%	78.93%	2.75%	4.52%
$P^\beta * R$	90.34%	66.24%	**48.37**%	**59.46**%

Table 2: Macro-averaged performance of the different scoring function on the development data. For $P^\beta * R$ we use β of 2 for the 25 most frequent languages in our experiment and β of 1 for the rest.

of their precision but only considers those with a precision higher than 90% performs comparably.

The best overall strategy is scoring phrases on their precision weighted recall with a variable β. Most importantly, this scoring function has the highest recall, even when subject to the Twitter rate limit. We argue that this is usually the objective of collecting Twitter data for a particular target language. For experiments on the test set, we use those lists of key phrases yielded by optimizing in this manner.

5.4. Test Set Performance

Table 3 shows the best greedy algorithm performance on the test set compared to the baselines. Even when selecting the best scoring function on the basis of development set results, it seems that the key phrases performed consistently. There is not big difference between their performance on the development set and the test set.

Method	Precision	Bound Recall	F-score
Baseline 1	14.11%	14.56%	2.64%
Baseline 2	89.58%	40.27%	51.51%
Greedy Selection	**90.38**%	**48.65**%	**59.71**%

Table 3: Performance of the baselines and suggested greedy selection algorithm on the test data.

The macro-averaged scores reported until now are useful in selecting the best general algorithm, but as can be seen in the full results in Appendix Table 6, there is a huge prevalence effect on individual target languages.

Even when accounting for the limit on number of tweets returned at any time, there is some variability in results between individual languages. We look at some of the outliers in detail in the next section.

6. Discussion

Performances for each of the target languages are recorded in Appendix Table 6, and show that while mostly consistent, some outlier results make it harder to discuss findings in a general way.

We mentioned the prevalence effect on recall earlier and thus focus on results that were unexpected with regards to language with similar frequencies on Twitter, specifically Chinese (zh), Esperanto (eo), Galician (gl) and Azerbaijani (az).

6.1. Confusion matrices

Table 4 shows the binary confusion matrices for four outlier results with the three most confused languages. Closer inspection of the confused tweets and selected key phrases give insight into two types of error.

First, for Chinese (zh), tokenization turned out to be a problem. We adopted the Twitter standard from (Twitter, 2019b), which is less suitable for logographic or abjad writing systems. For Japanese, Thai, Korean, Arabac and Hewbrew this turned out not to affect results in any noticeable way. Chinese gets confused often for these other languages however, and only a small portion of the target tweets is retrieved.

Esperanto (eo), Galician (gl) and Azerbaijani (az) all cope with another type of error. Their closeness to a more prevalent language (Spanish for Galician, Turkish for Azerbaijani and multiple highly frequent languages for Esperanto) forces the precision component in the greedy algorithm to select very rare occurrences. Although these phrases are successful in distinguishing between the target and their competition, their infrequency leads to a low recall for the target language in the test set.

	zh	other
ar	2,321	30,517
zh	1,673	3,321
ja	1,010	91,453
ko	228	25,088

(a) Chinese (zh)

	eo	other
eo	423	4,574
en	38	1,8M
es	22	426K
tr	13	81,219

(b) Esperanto (eo)

	gl	other
gl	2,016	2,961
es	1,268	753K
pt	633	453K
fr	102	161K

(c) Galician (gl)

	az	other
az	1,694	3,292
tr	36	583K
en	8	14M
fa	3	90,787

(d) Azerbaijani (az)

Table 4: Confusion matrices for target languages with subpar performances compared to other language with similar prevalence on Twitter.

For each of these outlier cases that bring down averaged performance, it would be interesting to see follow-up research that investigates how much improvement can be made, or whether the problem is with the data and possible code switching that occurs. Framing language identification on Twitter as a single-label problem introduces these inherent pitfalls.

6.2. Robustness and reproducibility

Although there are no major performance differences between applying the key phrase lists to the development and the test split of the data, there could be additional testing on the temporal nature of the lists. Training, development and tests were all performed on data yielded from the same six month snapshot, and could reflect specific events or topics of that period.

For example, the optimized key phrase lists contained 9 hashtags on average. Since hashtags are used mostly as topical and event markers, in a few years these search terms may have disappeared from Twitter completely.

Although this should lead to only marginally lower quality of the supplied phrases, it would be interesting to see an evaluation on data from another period. For now, the robustness of the method for selecting optimal key phrases is not under discussion. The code for generating key phrases on new Twitter snapshots and potentially new target languages is available at `https://github.com/tjkreutz/twitterphrases`.

7. Conclusion

We introduced a systemic way of selecting optimal key phrases for the the 50 most prevalent languages of Twitter. By demonstrating which tweets can be retrieved using the key phrases in an experimental setting that closely mirrors the setup with the real-time Twitter data stream, we provide the statistics that can be cited as limitations for Twitter collections built this way.

The best performing greedy algorithm for selecting key phrases, scores each phrase by precision weighted by recall. For the 25 most prevalent languages, exponentiating the precision with a β of 2 helps to increase the weight of high-precision phrases which limits the number of false positives in the resulting Twitter collection.

Alongside this paper and the code to generate new phrase lists, we provide all the lists as resources. Tracking Norwegian (no) tweets can be as simple as authenticating with your an API key and running curl:

```
    curl -d '@no.txt'
https://stream.twitter.com/1.1/statuses/filter.json
```

The resulting stream should consist of mostly Norwegian ($\pm$96%) language and make up more than half ($\pm$52%) of all available Norwegian tweets.

8. Bibliographical References

Barbaresi, A. (2016). Collection and indexing of tweets with a geographical focus. In *Proceedings of the 4th Workshop on Challenges in the Management of Large Corpora*, CMLC 2016, pages 24–27.

Barnaghi, P., Ghaffari, P., and Breslin, J. G. (2016). Opinion mining and sentiment polarity on twitter and correlation between events and sentiment. In *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 52–57. IEEE.

Basile, V. and Nissim, M. (2013). Sentiment analysis on italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, SSA 2013, pages 100–107.

Bergsma, S., McNamee, P., Bagdouri, M., Fink, C., and Wilson, T. (2012). Language identification for creating language-specific twitter collections. In *Proceedings of the second workshop on language in social media*, LSM 2012, pages 65–74. Association for Computational Linguistics.

Bloomberg. (2019). How twitter became ubiquitous in japan. `https://www.bloomberg.com/news/articles/2019-05-16/how-twitter-became-ubiquitous-in-japan`. Accessed: 2019-10-21.

Choudhary, N., Singh, R., Rao, V. A., and Shrivastava, M. (2018). Twitter corpus of resource-scarce languages for sentiment analysis and multilingual emoji prediction. In *Proceedings of the 27th International Conference on Computational Linguistics*, COLING 2018, pages 1570–1577.

Eisenstein, J., O'Connor, B., Smith, N. A., and Xing, E. P. (2010). A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1277–1287.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652.

Grave, E. (2017). Language identification. `https://fasttext.cc/blog/2017/10/02/blog-post.html`. Accessed: 2019-04-24.

Hong, L., Convertino, G., and Chi, E. H. (2011). Language matters in twitter: A large scale study. In *Fifth international AAAI conference on weblogs and social media*.

Huang, Y., Guo, D., Kasakoff, A., and Grieve, J. (2016). Understanding u.s. regional linguistic variation with twitter data analysis. *Computers, Environment and Urban Systems*, 59:244–255.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Kreutz, T. and Daelemans, W. (2019). How to optimize your twitter collection. *Computational Linguistics in the Netherlands Journal*, 9:55–66.

Laitinen, M., Lundberg, J., Levin, M., and Martins, R. (2018). The nordic tweet stream: A dynamic real-time monitor corpus of big and rich language data. In *Digital Humanities in the Nordic Countries 3rd Conference*, DHN2018, pages 349–362.

Leetaru, K., Wang, S., Cao, G., Padmanabhan, A., and

Shook, E. (2013). Mapping the global twitter heartbeat: The geography of twitter. *First Monday*, 18(5).

Petrović, S., Osborne, M., and Lavrenko, V. (2010). The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 25–26.

Praet, S., Daelemans, W., Kreutz, T., Van Aelst, P., Walgrave, S., and Martens, D. (2018). Issue communication by political parties on twitter. In *Data Science, Journalism & Media 2018, August 20, 2018, London, UK*, pages 1–8.

Scheffler, T. (2014). A german twitter snapshot. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC '14, pages 2284–2289.

Semiocast. (2011). Arabic highest growth on twitter english expression stabilizes below 40 `https://semiocast.com/publications/2011_11_24_Arabic_highest_growth_on_Twitter`. Accessed: 2019-10-12.

Tjong Kim Sang, E. and Van den Bosch, A. (2013). Dealing with big data: The case of twitter. *Computational Linguistics in the Netherlands Journal*, 3:121–134.

Twitter. (2019a). Developer agreement and policy. `https://developer.twitter.com/en/developer-terms/agreement-and-policy.html`. Accessed: 2019-06-20.

Twitter. (2019b). Filter realtime tweets. `https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter.html`. Accessed: 2019-06-25.

Twitter. (2019c). Standard search api. `https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html`. Accessed: 2019-06-25.

Language	ISO	FREQ	MEAN	MAX
English	en	39.06%	39.21%	46.93%
Japanese	ja	19.18%	19.09%	29.64%
Spanish	es	9.52%	9.45%	13.27%
Arabic	ar	7.29%	7.39%	10.82%
Portuguese	pt	5.17%	5.10%	9.59%
Korean	ko	4.43%	4.40%	6.78%
Thai	th	3.61%	3.58%	5.19%
Turkish	tr	2.05%	2.06%	4.04%
French	fr	1.88%	1.88%	3.55%
Chinese	zh	0.92%	0.94%	1.40%
German	de	0.88%	0.88%	1.14%
Indonesian	id	0.88%	0.88%	1.29%
Russian	ru	0.77%	0.78%	1.12%
Italian	it	0.61%	0.61%	0.96%
Telugu	tl	0.40%	0.40%	0.72%
Catalan	ca	0.39%	0.39%	0.68%
Hindi	hi	0.34%	0.34%	0.61%
Polish	pl	0.28%	0.28%	0.47%
Dutch	nl	0.26%	0.26%	0.38%
Persian	fa	0.22%	0.23%	0.42%
Malaysian	ms	0.16%	0.16%	0.23%
Egyptian Ar.	arz	0.15%	0.15%	0.28%
Urdu	ur	0.12%	0.12%	0.20%
Greek	el	0.12%	0.12%	0.20%
Esperanto	eo	0.10%	0.10%	0.10%
Finnish	fi	0.10%	0.10%	0.11%
Swedish	sv	0.09%	0.10%	0.14%
Bulgarian	bg	0.08%	0.07%	0.01%
Tamil	ta	0.07%	0.07%	0.13%
Ukrainian	uk	0.07%	0.07%	0.09%
Hungarian	hu	0.06%	0.06%	0.07%
Serbian	sr	0.06%	0.06%	0.09%
Galician	gl	0.05%	0.05%	0.08%
Cebuano	ceb	0.05%	0.05%	0.07%
Czech	cs	0.04%	0.04%	0.06%
Vietnamese	vi	0.03%	0.03%	0.05%
Kurdish	ckb	0.03%	0.03%	0.06%
Norwegian	no	0.03%	0.03%	0.03%
Danish	da	0.02%	0.02%	0.03%
Romanian	ro	0.02%	0.02%	0.03%
Hebrew	he	0.02%	0.02%	0.03%
Nepali	ne	0.02%	0.02%	0.03%
Bengali	bn	0.01%	0.01%	0.02%
Macedonian	mk	0.01%	0.01%	0.02%
Mongolian	mn	0.01%	0.01%	0.02%
Azerbaijani	az	0.01%	0.01%	0.02%
Marathi	mr	0.01%	0.01%	0.02%
Gujarati	gu	0.01%	0.01%	0.02%
Albanian	sq	0.01%	0.01%	0.01%
Kannada	kn	0.01%	0.01%	0.01%

Table 5: Language frequency (FREQ), averaged frequency per hour (MEAN) and maximum average frequency per hour (MAX) for the 50 languages in our data set.

Language	ISO	Precision	Bound Recall	F-score
English	en	40.21%	1.81%	3.46%
Japanese	ja	65.82%	2.96%	5.66%
Spanish	es	24.40%	2.18%	4.01%
Arabic	ar	80.03%	6.07%	11.28%
Portuguese	pt	89.36%	8.80%	16.03%
Korean	ko	97.73%	10.95%	19.70%
Thai	th	86.80%	11.20%	19.83%
Turkish	tr	94.64%	20.13%	33.19%
French	fr	95.65%	22.28%	36.15%
Chinese	zh	29.98%	3.64%	6.50%
German	de	91.44%	34.05%	49.62%
Indonesian	id	94.51%	39.04%	55.25%
Russian	ru	99.26%	56.17%	71.74%
Italian	it	93.75%	48.48%	63.91%
Telugu	tl	96.84%	81.02%	88.23%
Catalan	ca	97.74%	68.35%	80.44%
Hindi	hi	99.63%	97.86%	98.74%
Polish	pl	98.87%	59.60%	74.37%
Dutch	nl	98.25%	66.12%	79.04%
Persian	fa	99.36%	59.14%	74.15%
Malaysian	ms	93.45%	58.05%	71.62%
Egyptian Ar.	arz	99.78%	54.77%	70.73%
Urdu	ur	99.54%	87.52%	93.15%
Greek	el	99.69%	82.69%	90.39%
Esperanto	eo	81.03%	8.47%	15.33%
Finnish	fi	92.08%	27.70%	42.59%
Swedish	sv	97.42%	63.76%	77.07%
Bulgarian	bg	94.47%	72.51%	82.04%
Tamil	ta	99.80%	79.79%	88.68%
Ukranian	uk	94.62%	44.33%	60.38%
Hungarian	hu	88.78%	25.06%	39.09%
Serbian	sr	93.14%	58.11%	71.57%
Galician	gl	49.28%	8.67%	14.75%
Cebuano	ceb	89.63%	57.10%	69.76%
Czech	cs	98.06%	43.64%	60.40%
Vietnamese	vi	96.06%	76.45%	85.14%
Kurdish	ckb	99.51%	36.72%	53.64%
Norwegian	no	96.05%	51.92%	67.41%
Danish	da	97.14%	56.03%	71.07%
Romanian	ro	95.59%	52.53%	67.80%
Hebrew	he	99.95%	77.91%	87.56%
Nepali	ne	99.32%	88.09%	93.37%
Bengali	bn	99.94%	69.82%	82.21%
Macedonian	mk	99.01%	62.42%	76.57%
Mongolian	mn	99.83%	81.35%	89.65%
Azerbaijani	az	96.97%	33.98%	50.32%
Marathi	mr	97.87%	68.31%	80.46%
Gujarati	gu	99.60%	80.15%	88.82%
Albanian	sq	98.18%	64.01%	77.50%
Kannada	kn	98.72%	60.61%	75.11%

Table 6: Test set performance of individual target languages. In general less prevalent languages are easier to retrieve near-completely.