Computational Linguistics in the Netherlands Journal 9 (2019) 55–66 Submitted 09/2019; Published 12/2019

How to optimize your Twitter collection

Dutch keywords for better coverage

Tim Kreutz TIM.KREUTZ@UANTWERPEN.BE
Walter Daelemans WALTER.DAELEMANS@UANTWERPEN.BE

CLiPS, University of Antwerp

Abstract

Twitter allows API calls to retrieve one percent of all tweets at any time using a search word list. Since some languages, including Dutch, make up less than one percent of all tweets on average, a large part can be retrieved using the right keywords. This paper systematically assesses keyword lists for finding language-specific tweets. It contributes comparisons to previously suggested collection methods for the Dutch language and establishes the limitations of each. Generating keywords from Dutch tweets and picking 400 based on their precision-weighted recall achieves the overall best coverage at 91.3%. The list of Dutch keywords is made openly available alongside the code that can be used to generate lists for the collection of other languages or for other tasks that benefit from early filtering such as event or hate speech detection.

1. Introduction

Twitter data has been used for a wide range of research purposes. Tweets represent a huge textual resource that can be used for learning general language properties like word and n-gram frequencies (Bouma 2015, Gimenes and New 2016) and distributional information (Pennington et al. 2014). Twitter use is widespread, making its data suitable for work in dialectology (Eisenstein et al. 2010, Gonçalves and Sánchez 2014, Huang et al. 2016, Ljubešić et al. 2018) and language change (Eisenstein et al. 2014, Kulkarni et al. 2015). Organized challenges on popular NLP tasks such as opinion mining and author profiling benefit from the comparatively open nature of Twitter for creating shareable annotated datasets (for recent examples see Rosenthal et al. 2017 and Rangel et al. 2018 respectively).

Still, the rules on distributing tweets have recently become stricter (Twitter 2019a). It is only allowed to share tweet ids or abstract representations of their contents. Initiatives to share large general-purpose Twitter collections, such as the Edinburgh Twitter Corpus (2010) have been shut down. In recent years, the reality of using Twitter data in NLP is that most academic institutes maintain their own purposed collections.

Since not only Twitter collections, but also collection methods differ greatly, reproducibility is undermined. For example, data collected with the Search API is incomplete and inconsistent. Twitter states that: *"(...) the Search API is not meant to be an exhaustive source of Tweets. Not all Tweets will be indexed or made available via the search interface"* (Twitter 2019c). Additionally, different results may be displayed at different times. Especially when scholarly work is qualitative in nature and bases itself on smaller topic-based sets of tweets, undertaking the work at a different time or place can yield different results and conclusions.

Ideally, researchers would have access to an unchanging general-purpose collection that they can query retrospectively with their own methods. Even when such a collection is incomplete, knowing its sample size is preferred over querying black box APIs.

In this paper we provide a method for collecting language-specific Twitter datasets in real-time. We aim to address two problems set forth in this introduction. We address the dispersion of Twitter collections by publicly providing optimized and frequently updated lists of keywords for tapping language-specific Twitter data from the Streaming API. Anyone can use these lists to start their own language-specific tweet collection. Second, we address the problem that it is often unclear which limitations Twitter collections have. In the process of developing optimal lists that can be used to collect Dutch tweets, we provide the statistics that can be cited as limitations of the collection.

2. Previous work

Our method relies on the use of the Twitter Streaming API which filters and samples tweets in real-time. Bergsma et al. (2012) filter tweets by their geo-tags to create Twitter corpora for low-resource languages, and this yielded relatively pure collections for Arabic (99.9%) and Farsi (99.7%) but lower purity for Urdu (61.0%). Laitinen et al. (2018) use this type of filtering to create datasets for Nordic languages but they do not evaluate the purity of the data by hand. Since a very low percentage (between 0.7% and 2.9% depending on the country) of Twitter users enable location sharing, filtering by geo-tag yields very low coverage (Barbaresi 2016).

We alternatively aim to obtain the full collection of tweets for a language by optimizing the keyword filtering in the API. Our work is closest to Basile and Nissim (2013) (for Italian), Tjong Kim Sang and Van den Bosch (2013) (for Dutch) and Scheffler (2014) (for German). All of these papers use a list of target language keywords to filter the incoming stream of tweets. What differs across the papers, and this is where our contribution mainly lies, is the method to obtain these lists. Scheffler (2014) handpicked words from a list of most frequent German words. As they did not exhaustively filter cross-language homographs, the resulting collection of tweets contains a lot of other-language tweets. The approach in Tjong Kim Sang and Van den Bosch (2013) is similar but the list of frequent words is generated from Twitter data itself, resulting in Twitter-specific terms (such as hashtags) also being included in the final 229 keywords. The most principled approach to creating a list of keywords while limiting cross-language confusion is discussed in Basile and Nissim (2013). They created a list of frequent words from a large Italian corpus and exploited Google n-grams to rule out cross-language homographs. Using only the top 20 words for filtering resulted in a very precise collection (99.7% Italian) but no experiment was done to test the coverage.

Compared to the previous work we more closely inspect the Twitter API constraints to optimize keyword filtering. Specifically we consider phrases rather than single words, to make more precise selections in the stream. We generate these from a Twitter training set so that platform-specific phenomena (such as hashtags and mentions) are properly parsed and considered. We also compare various language-independent selection algorithms of these keywords and report the most important metrics for this problem, namely precision, recall and whether the method gets down-sampled by the API.

3. Twitter Streaming API constraints

We operate within the constraints of Twitter's Standard Streaming API, as opposed to the paid Enterprise edition. Our reasoning is twofold. First, we expect that most researchers do not have access to the paid editions, and would have to operate within the same constraints. Second, the keyword lists that we will generate as part of our method can also be used with the Enterprise API. Moreover, we show that using the lists is the best method to filter non-target language tweets, outperforming user based search and frequency-based keyword lists.

The most important constraint of the Standard API is that the search results at any second cannot amount to more than one percent of all tweets in that second (Twitter 2019b). If the rate limit is surpassed, the results are sampled down. This constraint puts a hard limit on the maximum recall of our keyword lists for certain languages. The upper bound will ultimately be negatively influenced by the general prevalence of a language on Twitter, and positively influenced by the competition of other languages at any time. Specific examples of the dynamics of language and time zones will be given in the following section.

Keywords	Will match	Will not match
(twitter,)	twitter, TWITTER, "twitter", #twitter, @twitter, `http://twitter.com`	newtwitter
(#twitter,), (@twitter,)	#twitter, @twitter	twitter
(twitter, api)	Twitter API, API Twitter	Twitter, API

Table 1: Demonstration of Twitter key words matching.

Besides rate limits, the parameters of the Streaming API are quite liberal. A user can filter the stream with 400 search terms. Terms need not contain single words, but can be any phrase of keywords up to 60 bytes. In the allowed UTF-8 encoding this would translate to 15 to 60 character search phrases depending on the character set of a language. These phrases are then parsed as conjunctive search tokens. The keywords "I am" would therefore match any tweet which contains "i" and "am" ignoring order and case in the tweet. Most punctuation in the tweet is also treated as separator, with exceptions made for hashtags (#) to signify topics and at (@) signs to refer to other users. The coverage of the full list then matches all tweets which contain one or more of the 400 key phrases. Some but not all of the subtle rules of Twitter's matching is available in the documentation (Twitter 2019b) but Figure 1 has been included for further clarity.

4. Data and processing

We use Twitter data from six months of tapping the Twitter Sprinkler, from October 2017 to March 2018. The Sprinkler randomly samples one percent of all tweets, which in the defined period resulted in 571,890,980 tweets. Although this may seem excessive, we will lose parts of this data in subsequent filtering steps. Moreover, the keyword lists are generated on Dutch tweets, which makes up only a small part of all Twitter data.

4.1 Labels and filters

Twitter assigns its own machine-detected IETF BCP 47 language tags. In addition to these off-the-shelf tags we consider the tags from the state-of-the-art fastText language identification model (Joulin et al. 2017). This model outputs 173 possible ISO language tags and a confidence score. Table 2 lists the agreement after conversion of the language codes between the two algorithms at different percentiles of fastText confidence scores. In the lower percentiles of confidence, agreement declines a lot. This is to be expected because FastText does more fine-grained language classification, whereas Twitter uses an 'undefined' label for languages its model cannot classify.

We filter the data by removing retweets and duplicate tweets from the data as shown in Figure 1. A labelling confidence filter is created on the basis of the fastText confidence at the 40th percentile, so that agreement between the two methods is at least 95.95 percent and the kappa score is at least 94.15. The latter filter hence removes 40% percent of the data. We argue that the use of

Percentile	Confidence	Agreement /w Twitter	Cohen's Kappa
0 - 10	>3.73%	30.24%	19.53
10 - 20	>32.41%	54.13%	40.93
20 - 30	>47.80%	74.78%	64.17
30 - 40	>62.93%	89.88%	85.01
40 - 50	>76.11%	95.95%	94.15
50 - 60	>85.69%	97.86%	97.11
60 - 70	>92.09%	98.79%	98.48
70 - 80	>96.24%	99.13%	98.97
80 - 90	>98.77%	99.46%	99.30
90 - 100	>99.84%	99.69%	99.53

Table 2: Agreement statistics between Twitter's method and the fastText language identification model.

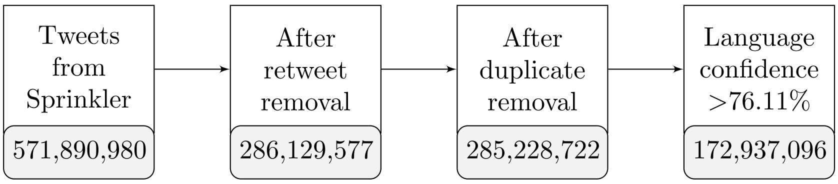

Figure 1: Retweets and duplicate tweets are removed from the data. In the last step, only tweets with high confidence of being labelled correctly by both the Twitter and fastText language classification are retained.

gold labels is not necessary here as long as we minimize the number of other-language tweets in a language-specific set.

4.2 Language prevalence

The timestamps of the tweets are used to see how languages relate to the constraints set out in Section 3. We calculate the average and maximum relative frequencies of present languages to give an idea of which languages can be collected completely, and if not, which part can likely be retrieved (Table 3). Only 14 languages surpass the 1 percent rate limit in Twitter in our data: Japanese, English, Arabic, Portuguese, Spanish, Korean, Thai, Turkish, French, Russian, Chinese, Italian, German and Indonesian. We highlight the relative frequency for the Dutch language during 24 hours in Figure 2. The average relative frequency of Dutch is around 0.28 percent, and the maximum relative frequency never surpasses 0.8 percent, which means that all Dutch tweets can theoretically be collected within the Twitter API constraints. Table 3 further shows the five languages with the highest average relative frequency, as well as the first five languages that can be collected completely.

4.3 Training procedure

To reiterate, our aim is to generate a list of Dutch keywords that retrieves as many Dutch tweets from the Twitter streaming API as possible, keeping in mind the 1% rate limit the API imposes.

The data is split into training data (80%), used for generating and selecting optimal keywords, development data (10%), used to evaluate suggested keyword selection methods, and test data (10%) to measure performance.

We use only the first month (October 2017) for training, since generating keywords is computationally expensive and providing updated lists generated over the previous month is our main focus.

National events or trending hashtags might influence the list of most distinctive Dutch phrases when generating them on data from a single month. We test the keyword list generated over data from October 2017 in a cross-month setting. The results realized in the same-month setting should diminish after some time if temporal artifacts play a role.

Language	**Avg. ↓**	**Max.**
Japanese	37.08%	66.01%
English	23.85%	38.41%
Arabic	10.37%	19.88%
Spanish	6.87%	19.01%
Portuguese	6.85%	19.27%
55 with lower averages.		
Language	**Avg.**	**Max. ↓**
14 with higher maximum averages.		
Persian	0.25%	0.93%
Polish	0.28%	0.88%
Dutch	0.28%	0.77%
Hindi	0.23%	0.75%
Catalan	0.07%	0.63%
41 with lower maximum averages.		

Table 3: Example languages sorted by average relative frequency and maximum relative frequency.

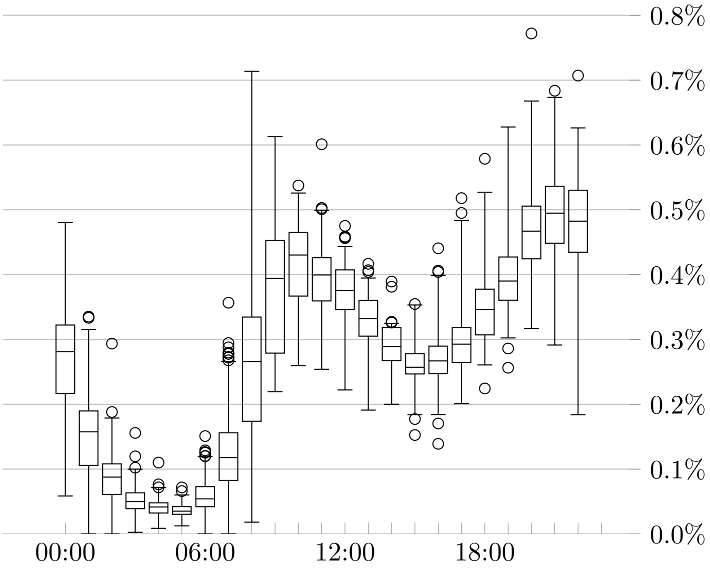

Figure 2: Average relative frequencies of Dutch tweets during 24 hours (GMT+2) over six months of data, grouped per hour.

5. Generating optimal keywords

Tweets are processed in accordance with Section 3: text is lowercased and all punctuation besides # and @ is treated as separators. Any combination of tokens in a tweet can be used as a query to collect that tweet. We thus generate token *powersets*; exhaustive combinations of tokens present in a tweet. Tweets are then indexed by their powersets.

What follows is a maximum coverage problem. Given a collection of keywords and the sets of tweets they identify, our objective is to cover the maximum number of Dutch tweets. Formally, we have a collection P of powersets, generated from a target language l, $P^l = \{P_1^l, P_2^l, ..., P_n^l\}$ and a parallel collection S of sets of tweets identified by those powersets, $S = \{S_1, S_2, ..., S_n\}$. We can select up to 400 sets from S to maximally cover a target language set of tweets T^l.

5.1 Greedy optimization

The list of candidate powersets is huge, so time and computational resources become a consideration in the optimization algorithm. We consider only the greedy algorithm, which is shown to be the best approximation algorithm in polynomial time (Feige 1998). The greedy algorithm picks the set with best coverage and removes the covered instances in its next iteration.

For a document retrieval problem, this approach would only optimize recall. The greedy algorithm selects powersets that cover many instances of the target language, regardless of whether they cover instances of other-language tweets. We thus experiment with a range of varieties of the algorithm that are listed below and further formalized in Figure 3.

5.2 Precision over recall

Instead of selecting the keywords with the highest coverage at each step, we select the keywords with the highest precision. In cases of a tie (usually at 100% precise input), the algorithm will consider recall.

5.3 Optimization with threshold

In both variations of the greedy optimization, whether optimizing recall or precision, a threshold is imposed on the other score. The algorithm still picks keywords that maximally cover the target set, but only under the condition that the result consists of at least 90% of the target language tweets. In the case where precision is valued most, the yielded tweets should at least amount to .25% (or $\frac{1}{400}$) of all target language tweets.

5.4 Precision-weighted optimization

The last variation weighs the coverage of keywords by their precision score. Instead of setting hard bounds on either precision or recall, an optimal score is seen as a balance of precision and recall. This approach should add more flexibility as hard thresholds may not be met after picking high-performance keywords from the candidates.

6. Results

The five objectives for the greedy algorithm select keywords with clear differences. In this section we compare their performance to each other and to existing baselines for selecting keywords.

```
Input  : $S$; $T$;
Output: Optimal keyword set $R$

Function Main($S$, $T$, $k \leftarrow 400$):
    $R \leftarrow \varnothing$
    for $i \leftarrow 0$ to $k$ do
        Remove tweets in R from every s in S.
        Add Optimal(S, T) to R.
    end
    return R

Function Optimal($S$, $T$):
    Select $S_i$ with best recall.
    OR
    Select $S_i$ with best precision.
    OR
    Select $S_i$ with best recall given a precision threshold
    OR
    Select $S_i$ with best precision given a recall threshold
    OR
    Select $S_i$ with best recall $\cdot$ precision.
    return $S_i$
```

Figure 3: Greedy algorithm: iteratively select a phrase s[i] from set S that best fits one of five objectives with regards to target set T and remove all found tweets by s[i] from the remaining items in S.

6.1 Optimization objectives

Table 4 shows the results of keywords generated on data from October 2017. If the number of returned tweets exceeds the 1% threshold that Twitter imposes, the result set is down-sampled to 1%. The down-sampling would affect the recall of the method being tested in a real-world setting; we report the corrected recall as *Bound Recall.*

Optimizing recall or precision without thresholds yields the lowest results. The former yields only unigrams with wide coverage over tweets that are not necessarily Dutch, such as: *co* (common in parsed URLS), *in* (yields 93% English tweets), *je* (yields 85% French tweets). The latter instead values longer powersets that retrieve only Dutch tweets, such as: *het in de* / it in the (covers 2.5% of Dutch tweets) or less frequent unigrams like: *zonder* / without (covers less than 1% of Dutch tweets). The other three objectives are more reasonable and all have their own advantages.

Valuing recall but imposing a 90% threshold on precision gives the best performance overall, without overstepping the upper bound on recall.

Valuing precision but imposing a minimum on retrieved tweets does not actually perform much worse in terms of precision than purely choosing the most precise keywords. This list is useful for obtaining Dutch tweets without needing to post-filter on language.

Objective	**Precision**	**Recall**	**Bound Recall**	**F1**
Optimize recall	0.007	**0.976**	0.024	0.011
Optimize precision	**0.999**	0.383	0.383	0.554
Precision threshold	0.991	0.885	0.885	**0.945**
Recall threshold	0.998	0.750	0.750	0.857
Precision-weighted recall	0.930	0.915	**0.915**	0.922

Table 4: Performance of keyword lists on unseen data generated under different optimization objectives. Twitter down-samples the stream of tweets when it exceeds the 1% rate limit. We report the down-sampled recall as *Bound Recall*.

The list that was optimized for precision-weighted recall is found to be the most suitable in the development setting. It achieves highest bound recall, and acceptable precision at 93%. We value recall slightly over precision since we might apply post-filtering using more sophisticated models in any case. Additionally, weighting recall by precision is a more general method that needs no tuning of threshold parameters to work. All further results use keyword lists generated by the precision-weighted recall method.

6.2 Alternative methods

We compare the generated keyword lists with baseline methods, a user-based method and a list proposed in previous work. We set up a random baseline to show what can be expected when indiscriminately tapping from the Twitter sprinkler. The second baseline is a list of the 400 most common words in Subtlex-NL (Keuleers et al. 2010).

Besides keyword-based search, Twitter allows user-based search, which taps tweets from the streaming API posted by a provided list of 5,000 user IDs (Twitter 2019b). For the user-based method we take the 5,000 users that tweeted out most Dutch tweets in our training data. We further compare to the list of 229 Dutch keywords that Tjong Kim Sang and Van den Bosch (2013) propose for the collection of Dutch Twitter data.

Table 5 shows that using frequent words has drawbacks similar to optimizing recall while generating keywords: using highly frequent (but not distinctively) Dutch words leads to imprecise performance. The user-based method retrieves a surprisingly large part of all tweets, showing that the 5,000 most active user accounts produce close to a third of all Dutch Twitter data.

The list of keywords composed by Tjong Kim Sang and Van den Bosch (2013) is based on frequent words and hashtags on Dutch Twitter. Care has been taken to remove words that are shared across languages. As such, the list does not confuse other-language tweets as much as the list from Subtlex. The number of retrieved tweets stays well below the 1% threshold, allowing the lower precision to be compensated by post-filtering of the Twitter collection.

The keyword list optimized for precision-weighted recall outperforms all other methods both in recall and precision. Using this list of keywords on the Twitter Streaming API would yield roughly 91.5% of all Dutch language tweets. Without post-filtering the collection would consist of approximately 92.2% of Dutch tweets.

Method	**Precision**	**Recall**	**Bound Recall**	**F1**
Random	0.003	0.010	0.010	0.005
Subtlex-NL	0.012	0.978	0.041	0.019
User-based	0.917	0.330	0.330	0.485
Tjong Kim Sang and Van den Bosch (2013)	0.576	0.901	0.901	0.703
Precision-weighted recall	**0.922**	**0.913**	**0.913**	**0.917**

Table 5: Comparison of baseline methods and state-of-the-art with suggested precision-weighted method.

6.3 Cross-month testing

We further experiment with the consistency of our optimized keyword list across time. National and local events taking place in a certain time frame, hashtag popularity and other temporal trends could change the keywords generated on data from a certain month. So far, we applied the keywords generated in October 2017 on test data from the same month. We test whether performance declines when applying the same list to months further down the line.

Figure 4 shows no clear decline when applying the keyword list in a cross-month setup. The best performance is achieved on test data from February 2018. The temporal aspect of keywords seems to mostly be ruled out by the importance of recall in the greedy optimization algorithm. Hashtags that appear among the selected keywords are more general such as: *#nieuws* / #news and *#nieuwstwitter* / #newstwitter.

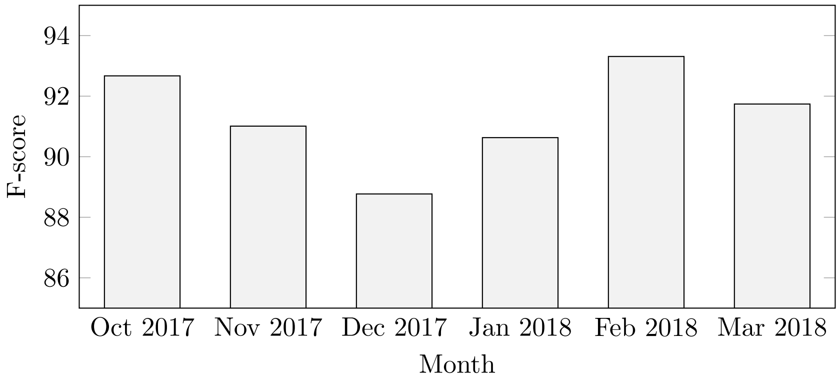

Figure 4: Cross-month performance of the keyword list generated for October 2017. The generated list does not contain temporal artifacts of the months that clearly affect performance when testing on other months.

7. Discussion

Some of the choices made in this research may warrant some closer inspection in future work. In section 4.1 we explain our decision to work with fastText language labels since it is the current state-of-the-art language identification model. Previous work had focused on obtaining Twitter data specifically to stimulate NLP research on low-resource languages. Although the fastText model obtains high results overall (including high agreement with the Twitter language identification model), specifically with low-resource languages in mind, it is a good idea to compare the labels we used with human expert annotations.

Since our data set consisted of six months of Twitter data, the finding that keyword list quality does not deteriorate with time can only be assessed for data at most six months apart. Further research can be done to see how temporal trends may influence keyword lists generated from data a few years apart.

8. Conclusion

The Twitter Streaming API can be used effectively for the near-complete collection of most languages in real-time. For the 14 languages that surpass the 1% rate limit the API imposes, the upper bound coverage depends on the busiest times of the day for each language.

We have investigated how to generate optimal keyword lists for querying the Twitter stream for Dutch tweets. By generating powersets of tokens over Dutch tweets, and iteratively picking 400 phrases based on their precision-weighted recall at each step, the highest coverage can be reached.

This approach compares favorably to methods that were suggested on troubleshooting websites and previous work, such as word lists based on frequent words (Scheffler 2014) or user-based collection. It also outperforms a list of distinctive Dutch words suggested in earlier work (Tjong Kim Sang and Van den Bosch 2013).

There are no clear differences in performance when testing the generated list of Dutch keywords in an in-month or in a cross-month setting. Although we hypothesized that event-specific keywords like hashtags could hurt the performance of optimal keywords over time, we found no evidence to support the claim. The hashtags that were present in the list generated over October 2017 are not event-related but rather general hashtags used frequently in Dutch tweets.

8.1 Provided optimal keyword lists

Strict rules are imposed on the distribution of Twitter data (Twitter 2019a). The reality of using Twitter data in NLP is that most academic institutes maintain their own collections. This may affect reproducibility when testing systems on data that has been collected in slightly different ways.

We share the optimal keyword list for collecting Dutch tweets.[1] Since Dutch tweets do not surpass the rate limit imposed by Twitter, any implementation of the keyword list for tapping the streaming API should yield the same collection of Dutch tweets.

Besides the benefit of comparability between datasets, the generated list was shown to yield a more complete set (91.3%) of Dutch tweets than previous methods. Without applying any post-filtering, the collection also predominantly consisted of Dutch tweets (92.2%).

8.2 Code for generating keywords

Beyond providing static lists, our approach can be adapted to fit any Twitter data collection goals. Primarily, using the same language detection approach, it can be fitted to any language, keeping in mind the API constraints on the larger languages.

But the same approach could theoretically be applied for other targets as well, as long as proper labels for the tweets are assigned. Any system that benefits from real-time detection, such as event or hate-speech detection systems, can tap the Twitter Stream using keywords indicative of the positive class.

We provide the basic code for generating keywords over self-supplied training data and running the greedy optimization algorithm to pick out optimal keywords in our Github repository.[2]

[1] `https://www.clips.uantwerpen.be/twitter/phrases/dutch-pwr`

[2] `https://github.com/tjkreutz/twitterphrases`

References

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

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

Bergsma, Shane, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson (2012), Language identification for creating language-specific twitter collections, *Proceedings of the second workshop on language in social media*, LSM 2012, Association for Computational Linguistics, pp. 65–74.

Bouma, Gosse (2015), N-gram frequencies for dutch twitter data, *Computational Linguistics in the Netherlands Journal* **5**, pp. 25–36.

Eisenstein, Jacob, Brendan O'Connor, Noah A. Smith, and Eric P. Xing (2010), A latent variable model for geographic lexical variation, *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pp. 1277–1287.

Eisenstein, Jacob, Brendan O'Connor, Noah A. Smith, and Eric P. Xing (2014), Diffusion of lexical change in social media, *PLOS ONE* **9**, pp. 1–13.

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

Gimenes, Manuel and Boris New (2016), Worldlex: Twitter and blog word frequencies for 66 languages, *Behavior research methods* **48** (3), pp. 963–972.

Gonçalves, Bruno and David Sánchez (2014), Crowdsourcing dialect characterization through twitter, *PLOS ONE* **9**, pp. 1–6.

Huang, Yuan, Diansheng Guo, Alice Kasakoff, and Jack Grieve (2016), Understanding u.s. regional linguistic variation with twitter data analysis, *Computers, Environment and Urban Systems* **59**, pp. 244–255.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov (2017), Bag of tricks for efficient text classification, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 427–431.

Keuleers, Emmanuel, Marc Brysbaert, and Boris New (2010), Subtlex-nl: A new measure for dutch word frequency based on film subtitles, *Behavior research methods* **42** (3), pp. 643–650, Springer.

Kulkarni, Vivek, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena (2015), Statistically significant detection of linguistic change, *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pp. 625–635.

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

Ljubešić, Nikola, Maja Miličević Petrović, and Tanja Samardžić (2018), Borders and boundaries in bosnian, croatian, montenegrin and serbian: Twitter data to the rescue, *Journal of Linguistic Geography* **6** (2), pp. 100—124.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014), Glove: Global vectors for word representation, *Empirical Methods in Natural Language Processing*, EMNLP '14, pp. 1532–1543.

Petrović, Saša, Miles Osborne, and Victor Lavrenko (2010), The edinburgh twitter corpus, *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, NAACL '10, pp. 25–26.

Rangel, Francisco, Paolo Rosso, Manuel Montes-y Gómez, Martin Potthast, and Benno Stein (2018), Overview of the 6th author profiling task at pan 2018: Multimodal gender identification in twitter, *CLEF 2018 Labs and Workshops, Notebook Papers*, CEUR Workshop Proceedings.

Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017), SemEval-2017 task 4: Sentiment analysis in twitter, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Association for Computational Linguistics, pp. 502–518.

Scheffler, Tatjana (2014), A german twitter snapshot., *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC '14, pp. 2284–2289.

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

Twitter (2019a), Developer agreement and policy. Accessed: 2019-06-20.

Twitter (2019b), Filter realtime tweets. Accessed: 2019-06-25.

Twitter (2019c), Standard search api. Accessed: 2019-06-25.