UNIVERSITÄT KOBLENZ · LANDAU
Faculty 4: Computer Science

WeST People and Knowledge Networks
Institute for Web Science and Technologies

Predicting Foreign Users from English conversations on Social Media

Bachelor's Thesis

in partial fulfillment of the requirements for
the degree of Bachelor of Science (B.Sc.)
in Informatik

submitted by
Alexander Winkens

First supervisor: Prof. Dr. Steffen Staab
Institute for Web Science and Technologies

Second supervisor: Ipek Baris
Institute for Web Science and Technologies

Koblenz, July 2020

Statement

I hereby certify that this thesis has been composed by me and is based on my own work, that I did not use any further resources than specified – in particular no references unmentioned in the reference section – and that I did not submit this thesis to another examination before. The paper submission is identical to the submitted electronic version.

	Yes	No
I agree to have this thesis published in the library.	☐	☐
I agree to have this thesis published on the Web.	☐	☐
The thesis text is available under a Creative Commons License (CC BY-SA 4.0).	☐	☐
The source code is available under a GNU General Public License (GPLv3).	☐	☐
The collected data is available under a Creative Commons License (CC BY-SA 4.0).	☐	☐

....

(Place, Date) (Signature)

Note

- If you would like us to contact you for the graduation ceremony,
  please provide your personal E-mail address: . . . .
- If you would like us to send you an invite to join the WeST Alumni
  and Members group on LinkedIn, please provide your LinkedIn ID : . . . .

Zusammenfassung

Social-Media Plattformen wie Twitter oder Reddit bieten Nutzern nahezu ohne Beschränkungen die Möglichkeit, ihre Meinungen über aktuelle Ereignisse zu veröffentlichen, diese mit anderen zu teilen und darüber zu diskutieren. Während die Mehrheit der Nutzer diese Plattformen nur als reines Diskussionsportal verwenden, gibt es jedoch Nutzergruppen, welche aktiv und gezielt versuchen, diese veröffentlichten Meinungen in ihrem Sinne zu beeinflußen bzw. zu manipulieren. Durch wiederholtes Verbreiten von bearbeiteten Fake-News oder stark polarisierenden Meinungen im gesamten politischen Spektrum können andere Nutzer beeinflußt, manipuliert und unter Umständen zum Träger von Hassreden und extremen politischen Positionen werden. Viele dieser Nutzergruppen sind vor allem in englischsprachigen Portalen anzutreffen, in denen sie sich überwiegend als Muttersprachler ausgeben. In dieser Arbeit stellen wir eine Methode vor, englische Muttersprachler und Nicht-Muttersprachler, die Englisch als Fremdsprache verwenden, anhand von ausgewählten englischen Social Media Texten zu unterscheiden. Dazu implementieren wir textmerkmalbasierte Modelle, welche für traditionelle Machine-Learning Prozesse und neuartigen AutoML-Pipelines zur Klassifizierung von Texten verwendet werden. Wir klassifizieren dabei Sprachfamilie, Muttersprache und Ursprung eines beliebigen englischen Textes. Die Modelle werden an einem bestehenden Datensatz von Reddit, welcher hauptsächlich aus englischen Texten von europäischen Nutzern besteht, und einem neu erstellten Twitter Datensatz, der Tweets von aktuellen Themen in verschiedenen Ländern enthält, angewandt. Wir evaluieren dabei vergleichsweise die erhaltenen Resultate unserer Pipeline zu traditionellen Maschinenlernprozessen zur Texterkennung anhand von Präzision, Genauigkeit und F1-Maßen der Vorhersagen. Wir vergleichen zudem die Ergebnisse auf Unterschiede der Sprachnutzung auf den unterschiedlichen Plattformen sowie den ausgewählten Themenbereichen. Dabei erzielen wir eine hohe Vorhersagewahrscheinlichkeit für alle gewählten Kategorien des erstellten Twitter Datensatzes und stellen unter anderem eine hohe Abweichung in Bezug auf die durchschnittliche Textlänge insbesondere bei Nutzern aus dem baltoslawischen Sprachraum fest.

Abstract

Social media platforms such as Twitter or Reddit allow users almost unrestricted access to publish their opinions on recent events or discuss trending topics. While the majority of users approach these platforms innocently, some groups have set their mind on spreading misinformation and influencing or manipulating public opinion. These groups disguise as native users from various countries to spread frequently manufactured articles, strong polarizing opinions in the political spectrum and possibly become providers of hate-speech or extremely political positions. This thesis aims to implement an AutoML pipeline for identifying second language speakers from English social media texts. We investigate style differences of text in different topics and across the platforms Reddit and Twitter, and analyse linguistic features. We employ feature-based models with datasets from Reddit, which include mostly English conversation from European users, and Twitter, which was newly created by collecting English tweets from selected trending topics in different countries. The pipeline classifies language family, native language and origin (Native or non-Native English speakers) of a given textual input. We evaluate the resulting classifications by comparing prediction accuracy, precision and F1 scores of our classification pipeline to traditional machine learning processes. Lastly, we compare the results from each dataset and find differences in language use for topics and platforms. We obtained high prediction accuracy for all categories on the Twitter dataset and observed high variance in features such as average text length especially for Balto-Slavic countries.

Contents

1. Introduction

Social media has become an integral part of today's society, ranging from casual conversation to serious discussions and debates. It is also one of the largest medium for spreading opinions, especially by few, large influencers and their followers. The more followers or retweets a user has on a specific topic, the more influence they will have on public opinion [Cano et al., 2014]; users therefore "act as proxy of topical influence by means of retweet relations". While most users approach the platforms innocently and merely wish to keep up with the current events, some take advantage of the openness and anonymity by e.g. creating dummy accounts which spread misinformation or content targeted to specific user groups, in an attempt to influence public opinion on controversial topics. One well known example is the presidential election debates in the USA between Donald Trump and Hillary Clinton in 2016 [Ghanem et al., 2019], which was riddled with content posted by Russian bots. To identify the origin of a post we analyse language semantics, syntax and topical context and find similarities in usage for non-Native English speakers of different countries and nationalities.

Writing behaviour varies drastically for different demographics including nationality, gender, age and personality with the majority of Twitter users being under or around 20 years and evenly split between genders [Nguyen et al., 2013]. Females tend to use more emotional words and first-person singulars, while also mentioning more psychological and social processes. In contrast, males use more swear swords and object references [Schwartz et al., 2013]. While younger users (aged 13 to 18) stick to school related topics and 'Internet speak/slang', this slowly transitions to college and the 'drunk' topic for ages 19 to 22. The trend from school to college and work also shows a decrease in the usage of 'I' and an increase in 'We', indicating the "importance of friendships and relationships as people age". Extroverts mention social words more frequently (e.g. 'party', 'boys', 'ladies'), while introverts stick to solitary activities ('computer', 'reading', etc.) and are more interested in Japanese media (e.g. 'anime' and 'manga'). Also, emotionally stable users are more vocal about enjoyable social activities such as 'sports', 'vacation' and 'family time'. Users change their reply behaviour for different topics (e.g. a users reply to a political debate show different emotions than to a new technology) [Kim et al., 2012], which brings a change in linguistics with different emotional states [Chen et al., 2010].

We provide a way to identify user nationality from both West and East based on their generated content. Using linguistic features such as Parts-of-Speech (e.g. usage of nouns, adverbs etc.) tailored for better recognition of modern slang and abbreviations, spelling/grammar mistakes and word frequency, we discern different languages and build language/feature models. We train and evaluate models with a Reddit corpus, which already includes labelled data for languages and domains, and weakly annotated data from Twitter (by investigating other content such as recent tweets and profile information to assume a matching country-of-origin) to increase language and topic coverage. We hypothesise that language differs more severely on Twitter due to the character limitation and openness of discussion compared to a more traditional forum-like approach on Reddit. Based on these considerations, we develop three research questions that are answered in this thesis. i) How strongly does text style differ cross-platform and among different domains? ii) Can native language identification from English text solely based on linguistic features obtain accurate results? iii) How does an automated machine learning pipeline perform compared to basic classification models on the tasks of language identification?

The contributions of the thesis are summarised as follow:

1. We collect a dataset from selected topics and trending hashtags on Twitter by extracting tweets from the categories Arts/Culture, Business/Technology/Science, Politics and Social/Society.

2. Our pipeline classifies a total of 19 different languages in four language families for Reddit from European and non-European domains, and eight different native languages in four language families and categories for Twitter. We extract text features from the Reddit and Twitter dataset such as word and character n-grams, Parts-of-Speech tokens and text length. We implement AutoML (automated machine learning) pipelines which take these features as input for predicting origin, native language and language-family as shown in Figure 1.

3. We evaluate the performance of the pipelines by comparing the prediction results to basic classifiers such as Random Forest and a baseline score elevated from the works of Goldin et al. [Goldin et al., 2018].

4. We obtain over 94% accuracy in predicting Native and non-Native English speaking users on Twitter, over 85% correct predictions for language family, over 66% for native language and 82% for categories. Our pipeline also scored 34% prediction accuracy for native language identification on the Reddit dataset.

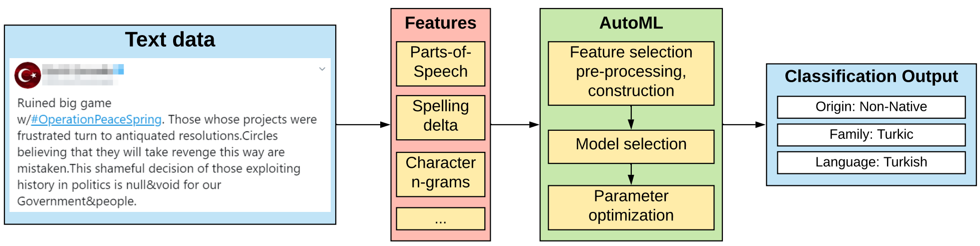

Figure 1: Classification Framework

The structure of the thesis is as follows.

Section 2 introduces background and related works, such as the original work by Goldin et al. and Volkova et al. [Volkova et al., 2018].

Section 3 describes the structure of the Reddit and Twitter datasets, and explains their characteristics as well as how they were pre-processed for usage in our models. We also introduce our feature-set, how each feature is created and the reasoning for using it.

In section 4 we highlight research tasks and present the setup and methods for the experiments. Results for each implementation and dataset are discussed and we observe which pipeline had the highest success in native language, language-family and origin prediction.

In Section 5 we evaluate the classification results for each dataset and pipeline. We also present language differences by examining feature data.

Section 6 discusses thoughts on our work and some possible improvements on data collection and methods.

2. Related works

2.1. Background

The baseline for this work is introduced in Goldin et al., which dealt with the problem of identifying 23 native languages on data extracted from *Reddit*. Their process consisted of three parts. i) to distinguish between Native and non-Native authors, ii) to determine the language family (e.g. Germanic or Romance), iii) to identify the native language of non-Native authors. Native and non-Native users were distinguished by using the metadata *flairs* from Reddit, which allows users to tag themselves with e.g. their country. Countries with the same official language (e.g. Germany and Austria) were combined, even though they may have slight differences in their language style. Additional to basic features such as Parts-of-Speech and sentence length, they also employed content based features (e.g. token n-grams and character n-grams) and spelling/grammar errors. Their results were at highest 86% prediction accuracy for in-domain (only European sub-reddits), and 79% for out-of-domain (only non-European sub-reddits) datasets.

Similar tests were made on a language family classification task on the same dataset [Rabinovich et al., 2018]. Instead of comparing stylistic features, frequencies of unbiased words which they expect to be distributed differently based on synonyms with divergent etymologies were weighed. For their work they eliminated cultural bias from the data (e.g. country-specific contextual language such as wine in France, beer in Germany etc.) by finding words that were overused in certain countries. They also calculated a distance between two English texts based on the frequency of a given word in both texts and a vector representation of the author, which includes 'information about a subject' such as context for word usage (e.g. *wicked* is used differently in the USA than in the United Kingdom).

In [Volkova et al., 2018], various linguistic features such as Parts-of-Speech tokens were manually gained from over one million tweets of different non-Native English speakers to create a model for identifying second language users. This was done by "state-of-the-art machine learning models trained on lexical, syntactic, and stylistic signals learned from word, character and byte representations extracted from English only tweets" dissecting tweets into their basic components such as number of URLs, hashtags, emojis, usage of punctuation and word elongation, or number of verbs, nouns etc. used. While they offer a lower language quantity compared to the Reddit corpus of Goldin et al., their data also includes Asian and Austronesian countries.

Performance of language identification algorithms when applied to tweets with transliterated text was studied in [Cardoso and Roy, 2016]. Their work includes Russian and Arabic transliteration (e.g. no access to a Cyrillic typeset and write Russian words in the Latin alphabet) and their effect on prediction accuracy. As most of these transliterations appear like typographical errors they assumed it would negatively impact performance. The language classification process found in

[Lui and Baldwin, 2012] was implemented and extended with Arabic and Russian transliteration. It was compared to the original version in four different corpora: personal sources such as blogs, forums and communities, professional sources from newspapers and government pages, micro-blogging sources such as Twitter, and comments from social sites such as Youtube and Facebook. The model was trained on short, noisy data and resulted in higher accuracy for micro-blogging sites compared to the original process. A similar model which was considering transliterated text resulted in lower performance over-all.

A different approach for language identification is the usage of user profiling. It relies on building user profiles from platform specific features. [Eke et al., 2019] specifies different State-of-the-Art processes for various data sources e.g. Twitter. Their Twitter profiling consists of features such as *User interest*, *Number of friends* and *tie strength* between users and their friends. Instead of relying solely on linguistic features, it focuses on social features and information gained by the users profiles and connections. However, for native language identification it relies on voluntary self-labeling by the users as it extracts user locations from their Twitter profile. Users may not disclose their native country and use their current location or leave it blank instead. Another technique is investigating user engagement by focusing on features such as tweets, tweets by followees and Twitter metrics such as *retweets* and *likes*. This generates a node-map based on user interest and highlights like-minded users.

Similarly to the works of Goldin et al. and Volkova et al. we use linguistic features to identify non-Native English speakers. For Reddit, we classify a total of 19 different languages in four language families, both from European and non-European sources. We create a new dataset for Twitter based on hashtags and topics instead of user-profiling for the four language families Indo-European, Indo-Aryan, Japonic and Turkic. We implement a feature to calculate similarity between n-grams and compare it to the performance of term frequency inverse document frequency.

2.2. Automated Machine Learning

Automated Machine Learning (or **AutoML**) aims at automating machine learning processes, especially hyperparameter tuning, to assist in finding optimal parameters and settings for various models and/or datasets. Areas which are targets of automation are *Data preparation* (e.g. detection of data types and intent, task detection), *Feature engineering* (feature selection, extraction, detection and handling of missing data, transfer learning), *Model selection*, *Hyperparameter optimization*, *Pipeline selection* with various constraints such as memory, time and complexity, *Evaluation selection* (metrics and validation methods used for evaluating the predictions), *Problem detection* and *Result analysis*.

AutoML attempts to replace the human component in each of these areas (such as manually designing and constructing features from a given dataset) by automating these processes. It also aims at being a generalised tool for machine learning i.e. it can be used on any input data and learning task without any further modifications.

Core goals of AutoML are defined by [Yao et al., 2018] as i) Good performance: good generalization performance across various input data and learning tasks can be achieved, ii) Less assistance from humans: configurations can be automatically done for machine learning tools, and iii) High computational efficiency: the program can return a reasonable output within a limited budget. To achieve these goals, AutoML uses a basic optimiser and evaluator framework. The evaluator measures the performance of a model and its hyperparameter setup on a given dataset; The optimiser manages hyperparameter and model selection for the process. Output from AutoML pipelines are learning tools used for classification tasks. This process is usually done by manually trying a configuration and evaluating the resulting feedback, which in case of AutoML is all done automatically.

In [He et al., 2019] various methods for automation are introduced in those areas. Data collection generally is a very tedious and time consuming step of the pipeline as each piece of data has to be analysed and labelled manually. Automating the dataset creation is something that would drastically reduce the time spent on the classification pipeline. Methods such as creating a strong labelled sample dataset, comparing various other data to this sample and clustering closely related ones are a part of these automation processes. Others include offsetting dataset imbalance by creating synthesised samples between different minority-samples instead of up or down-scaling the dataset.

Tree-based Pipeline Optimization is a part of automated machine learning and aims at automating three steps of common machine learning pipelines: i) Feature selection, pre-processing and construction, ii) Model selection and iii) Parameter optimization. [Olson et al., 2016] have shown that their *Tree-based Pipeline Optimization Tool* (or **TPOT**) finds pipelines which consistently offer the same accuracy as guided pipelines with 'little to no input nor prior knowledge from the user'. It employs algorithms from the commonly used *scikit-learn* [Pedregosa et al., 2011] but also efficient and powerful methods such as *Extreme Gradient Boosting*. This can help in making machine learning more accessible and creating baseline pipelines providing good results, while avoiding mistakes such as over or underfitting. However, they also show that finding these randomly generated pipelines tends to be slower, especially for larger datasets which can take several hours and requires high computational power.

3. Methodology

This section gives an in-depth overview of the datasets as well as a general overview of their platform structure. First, we describe how Rabinovich et al. obtained and annotated the Reddit dataset [Rabinovich et al., 2018] which is used in this work. We explain the data acquisition and annotation process for the Twitter dataset, and the methods of pre-processing the data to reduce the overall text bloat. We introduce the features and how each feature was implemented as well as categorizations for each dataset. Lastly, we implement the models used for the classification: *Random Forest*, *Logistic Regression*, *Support Vector Machine* and the *TPOT* pipelines.

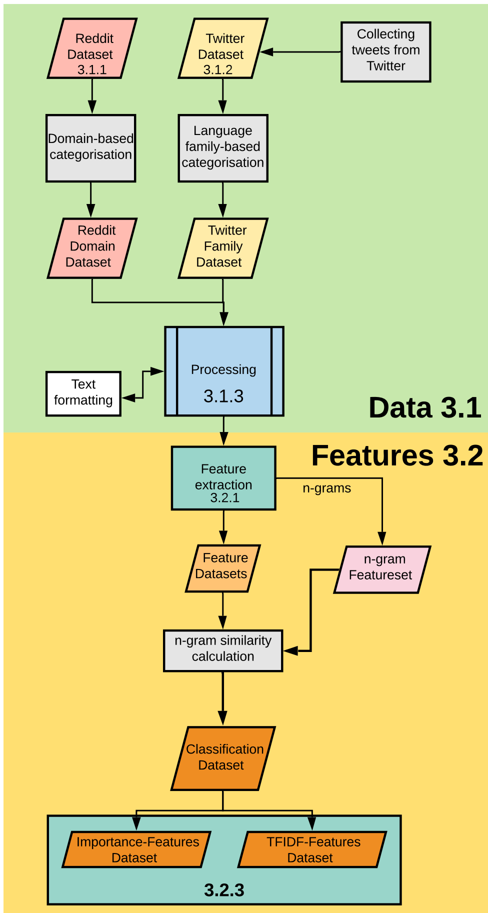

Figure 2: Process from raw datasets to feature datasets used in classification tasks

3.1. Data

In the following section we will describe the acquisition and pre-processing of the two datasets, a pre-labelled dataset from *Reddit*, and a newly created dataset from *Twitter*. First, we describe the Reddit platform and the dataset structure. Next we give details on Twitter and the process of creating dataset such as finding suitable data sources and annotation.

3.1.1. Reddit

Reddit is one of the largest social media/news sites with over 330 million active users[1] monthly. The site functions solely on user-generated content, or posts, which can either be up-voted, to increase traffic and popularity, or down-voted with the opposite effect.

Posts are specific to so called sub-reddits, which are topical categories such as Politics, News and more nuanced topics such as specific sports-clubs, cities or events. Sub-reddits can be freely created and moderated by the users which is in line with the hands-off approach of user-generated content. Posts can be links to other sites, media such as images or videos, or simple text posts. Users can comment and discuss on each of these posts. Comments can also be up -and downvoted, with the highest up-voted comments displayed at the top by default. Each comment generates a sub-post, to which users can respond and create a comment-chain.

We used the dataset from [Rabinovich et al., 2018] for comparison with the Twitter dataset. This dataset was created by extracting posts and comments from sub-reddits with users who specify their native country as so called flairs. These include *Europe*, *AskEurope*, *EuropeanCulture*, *EuropeanFederalists* and *Eurosceptics*. From these sources over nine million posts by 45.000 distinct users were annotated and used as a *seed corpus*.

As the user comment history is public and users were already associated with a country, Rabinovich et al. extracted all other comments to create the final dataset of over 250 million sentences in 80.000 different sub-reddits. After removal of multilingual countries and countries with less than 500.000 total posts, random samples were grouped into '(i) Native vs. non-Native English speakers, (ii) the three Indo-European language families, and (iii) 45 individual native languages'. Function words and Parts-of-Speech tri-grams were created for each group and used for classifying.

Rabinovich et al. obtained 90.8%, 85.2% and 60.8% prediction accuracy for the three groups respectively, giving flairs a reputable way to identify a users native country. Trimming the dataset by '(i) removing text by users who changed their country flair within their period of activity; (ii) excluding non-English sentences; and (iii) eliminating sentences containing single non-alphabetic tokens' formed the final dataset of over 230 million sentences, which was used for our purposes.

---

[1] As of 2018

The dataset[2] consists of several language files separated into Native and non-Native speakers, which are further categorised into data from European and non-European *sub-reddits*. Each file contains texts, usernames and the sub-reddits (see Figure 3) in which they were posted in. The labeling was done by cross-referencing [Rabinovich et al., 2018] usernames with a thread in which users posted their native countries. From these, posts were extracted from users which were deemed as highly likely to be of specific countries. In total 25 countries were included: *Australia*, *Ireland*, *New Zealand*, *United Kingdom*, *United States*, *Bulgaria*, *Croatia*, *Czech*, *Lithuania*, *Poland*, *Russia*, *Serbia*, *Slovenia*, *Austria*, *Finland*, *Germany*, *Netherlands*, *Norway*, *Sweden*, *France*, *Italy*, *Mexico*, *Portugal*, *Romania*, and *Spain*. Countries with the same official language were combined into a single one (e.g. Germany and Austria, Spain and Mexico).

```
12341                                  [user]

europe                                 [subreddit]

& gt ; Yet , thousands of people risk their lives
crossing the seas in order to reach that
horrible place that is the EU.\\n\\nAnd
a good number want to get to the UK .   [post]
```

Figure 3: Sample from European sub-reddit data by American users. Usernames are unidentifiable.

[2] http://cl.haifa.ac.il/projects/L2/index.shtml

To create a common baseline, we divided the data into categories similar to those found in Rabinovich et al.:

Language family	Included countries
Native	Australia, Ireland, New Zealand, United Kingdom, United States
Romance	France, Italy, Mexico, Portugal, Romania, Spain
Germanic	Austria, Finland, Germany, Netherlands, Norway, Sweden
Balto-Slavic	Bulgaria, Croatia, Czech, Lithuania, Poland, Russia, Serbia, Slovenia

Table 1: Language family and country categorisation for Reddit

Country	Number of posts in European sub-reddits	Number of posts in non-European sub-reddits	Percentage of posts in language family	Percentage of total posts
Australia	10882	1649571	4.64%	2.36%
Ireland	67191	3680080	10.47%	5.33%
New Zealand	2284	378688	1.06%	0.54%
United Kingdom	224004	13086173	37.18%	18.93%
United States	146962	16552221	46.65%	23.75%
Bulgaria	27390	475030	8.96%	0.71%
Croatia	26764	552801	10.34%	0.82%
Czech	36738	694144	13.03%	1.04%
Lithuania	30116	515310	9.73%	0.78%
Poland	112867	1714414	32.59%	2.60%
Russia	31167	586398	11.01%	0.88%
Serbia	24876	452238	8.51%	0.68%
Slovenia	25660	301189	5.83%	0.46%
Austria	42797	1056080	5.84%	1.56%
Finland	64153	2145515	11.74%	3.14%
Germany	224262	5658306	31.24%	8.37%
Netherlands	122403	4774382	26.01%	6.97%
Norway	31889	1522319	8.26%	2.21%
Sweden	68738	3116496	16.92%	4.53%
France	89768	2164168	30.15%	3.21%
Italy	44188	986925	13.79%	1.47%
Mexico	1869	238656	3.22%	0.34%
Portugal	47441	1327155	18.39%	1.96%
Romania	74958	1100886	15.73%	1.67%
Spain	65084	1333932	18.72%	1.99%
Total	1796167	68505191		

Table 2: Number of posts in Reddit dataset by language

The dataset is imbalanced due to the high quantity of native English posts (see Table 2), which almost make up 50% of the total (e.g. 23.75% of the total posts are made by users from the United States, compared to 0.34% made by Mexican users). We sampled 10000 posts from each language family with equal distribution for languages, equalling to 5% of the lowest language post count and 0.06% of the highest. Each post is labelled with their native language, language family and either Native or non-Native origin. We create two distinct datasets for European and non-European posts.

3.1.2. Twitter

Twitter is a micro-blogging site with over 321 million active users[3]. Its main differences compared to other blogging sites were *hashtags* for creating discussion topics, the 140 (280 since November 2017) character limitation on each tweet/post, and ability to follow certain individuals for updates. In general, Twitter is used for casual conversations similar to SMS, and open, fast-paced discussion on trending news or events.

Recently, Twitter has garnered more and more criticism for allowing the spread of misleading information and hate-speech. Especially during the early stages of the COVID19 outbreak, many users were flagged and/or removed for misconduct due to spreading incorrect information. This led to Twitter tagging[4] posts as *misleading*, *disputed* or *unverified*, which was also target of criticism as people were concerned about limiting their freedom of speech.

The tagging system has since been broadened to include tags such as *public interest notice* and *glorified violence*, which was especially used during the George Floyd protests in May, 2020, to stop the spread of hate-speech[5].

3.1.2.1. Limitations

To create our Twitter dataset we collect tweets from various hashtags. Even though Twitter offers an API to developers which allows the extraction of tweets with text and metadata, the lowest (free) access level limits the amount of requests for searching tweets to 250 (with 100 tweets per request) every month. Additionally, only tweets from the last 30 days are available with the API. The limit for access to the full Twitter archive is 50 per month. To increase the limits, a premium subscription is required. However, even the most expensive option does not include tweets which are more than 30 days old. Since our method uses trend-based hashtags which could be up to five years old, the standard API would not work.

3.1.2.2. Circumvention with NASTY

Nasty Advanced Search Tweet Yielder[6] is a tool to query Twitter and extract query results. Instead of using the limited Twitter developer API, it simulates a normal web browser accessing the Twitter website. This allows access to the full tweet archive, ranging all the way back to 2006, better filtering options and no request-limitation.

The author states that NASTY technically violates the Twitter Terms-of-Service as it does not conform to the permission rules set by Twitter. Using the tool itself is still legal as 'It is unclear (and dependent on jurisdiction) to whom the TOS apply. Since using NASTY does not require signing in to Twitter or opening it manually

[3] As of February 2019

[4] https://blog.twitter.com/en_us/topics/product/2020/updating-our-approach-to-misleading-information.html

[5] https://www.bbc.com/news/technology-52846679

[6] https://github.com/lschmelzeisen/nasty

in a web browser, a court may decide that the user never agreed to the TOS and is therefore not bound to its conditions'. Also, 'in Germany up to 75% of any publicly accessible database (here, Twitter) may [be] copied for academic research'[7]. Since the aforementioned does not imply that sharing the dataset is legal, we will not be making the original Twitter dataset available, but instead share the dataset without any user identification.

3.1.2.3. Collecting data

For the Twitter dataset we first had to find suitable sources to gather tweets. We investigated various *hashtags* that were trending in at most one country at a specific timestamp (e.g. #ExtinctionRebellion is a political hashtag originating in England but was trending only in Germany on November, 20th). We also suggested different categories which we assume to have the most variance in both topicality and technicality: *Arts/Culture*, *Business/Technology/Science*, *Social/Society* and *Politics*. At least two trending hashtags were used for each category (with one exception due to the quantity of tweets as seen in Table 4) and language. Due to the limited amount of trending hashtags in foreign countries with English text we minimised the scope to countries similar to the ones in the Reddit dataset and non-European countries like Japan and India.

Language family	Included countries
Turkic	Turkey
Indo-European	France, Greek, Germany, Russia
Japonic	Japan
Indo-Aryan	India
Native	English

Table 3: Language family and country categorisation for Twitter

[7]https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3491192

Country	Arts/Culture	Business/Technology/Science	Social/Society	Politics
Turkey	#SezenAksu #Cemre #BugünGünlerdenGALATASARAY #BugünGünlerdenTrabzonspor	#Teknofest2019 #coronaviruesue	#Perşembe #salı	#BaharKalkanı #DünyanınEnGüçlüOrdusuyuz
France	#MariesAuPremierRegard #JeudiPhoto	#CoronavirusFrance #ChangeNOW2020	#negrophile4life #JeSuisVictime #CesarDeLaHonte	#49al3 #greve20fevrier
Greece	#AdinamosKrikosGr #tokafetisxaras #paokoly	#mitefgreece #reloadgreece	#Τσικνοπεμπτη #28ηΟκτωβριου	#εβρος #μεταναστες
Germany	#AUTGER #DerSchwarzeSchwan	#spiegelonline #BahnCard	#Umweltsau #Weltknuddeltag	#Sterbehilfe #Bauernproteste #dieUhrtickt
Russia	#BTSTOUR2020_RUSSIA #Биатлон	-	-	-
Japan	#popjwave #annkw	-	-	-
India	#PonniyinSelvan #NewEra_By_SaintRampalJi	#IISF2019	#AskSaiTej #Dabangg3Reviews	#99535_88585_AgainstCAA #AzadiForAzad
America	#titansvschiefs #winniethepoohday	#SAMESBC #ngcx	#NationalDressUpYourPetDay #ThingsThatUniteUs	#TellTheTruthJoe #VirginiaRally
Worldwide	#GameOfThrones #BoyWithLuv	#CES #COVID2019	#loveyourpetday #2020NewYear	#Hanau #InternationalWomensDay

Table 4: List of hashtags in each country and category. Hashtags for foreign languages may not be in English and for some categories no suitable hashtags were found.

We used NASTY to query Twitter with each hashtag, the Twitter specific *English* parameter, and timestamp (if needed) to extract up to 1000 matching tweets. The extracted dataset contains the text, user and profile information, the tweet-URL, timestamp and various metrics such as retweets and likes. We first filter any non-English text with *PolyGlot* [8]. From the resulting tweet data, we manually remove spam and any possible leftover non-English tweets. We manually check user profiles and post history for each tweet to identify hints pointing to their native language (e.g. Twitter biography, other posts containing their native language, links helping in identification such as personal blogs or websites, or engagement in country specific hashtags/discussions). Tweets which still raised doubts either due to not being able to discern their language from other similar languages (e.g. Russian and Ukrainian) or not enough conclusive data were not included in the final dataset.

[8]https://github.com/aboSamoor/polyglot

Germany	
#AUTGER	99, 93, 76
#DerSchwarzeSchwan	111, 100, 88
#spiegelonline	189, 168, 62
#BahnCard	80, 64, 40
#Umweltsau	96, 50, 41
#Weltknuddeltag	113, 69, 57
#Sterbehilfe	39, 21, 11
#Bauernproteste	55, 47, 27
#dieUhrtickt	23, 21, 15

France	
#MariesAuPremierRegard	26, 22, 17
#JeudiPhoto	63, 61, 42
#CoronavirusFrance	242, 235, 52
#ChangeNOW2020	105, 105, 44
#negrophile4life	47, 39, 16
#JeSuisVictime	113, 84, 34
#CesarDeLaHonte	48, 46, 21
#49al3	177, 157, 59
#greve20fevrier	66, 62, 13

Greece	
#AdinamosKrikosGr	106, 96, 73
#tokafetisxaras	53, 59, 38
#paokoly	66, 65, 52
#mitefgreece	63, 63, 61
#reloadgreece	96, 96, 90
#Τσικνοπεμπτη	95, 85, 67
#28ηΟκτωβριου	79, 75, 64
#εβρος	199, 195, 128
#μεταναστες	60, 48, 32

India	
#NewEra_By_SaintRampalJi	409, 409, 407
#PonniyinSelvan	255, 240, 236
#IISF2019	98, 98, 98
#99535_88585_AgainstCAA	74, 74, 74
#AzadiForAzad	124, 122, 116
#AskSaiTej	63, 59, 55
#Dabangg3Reviews	77, 76, 76

Japan	
#popjwave	95, 82, 81
#annkw	121, 99, 83

Russia	
#Биатлон	111, 97, 75
#BTSTOUR2020_Russia	78, 73, 71

Turkey	
#DünyanınEnGüçlüOrdusuyuz	64, 54, 39
#Cemre	120, 105, 91
#BugünGünlerdenGALATASARAY	234, 145, 119
#BugünGünlerdenTrabzonspor	75, 33, 29
#Teknofest2019	95, 90, 82
#coronaviruesue	79, 77, 53
#Perşembe	169, 130, 102
#salı	259, 226, 141
#BaharKalkanı	190, 182, 133

Native	
#titansvschiefs	95, 95, 95
#winniethepoohday	96, 96, 96
#SAMESBC	97, 97, 97
#ngcx	101, 101, 101
#NationalDressUpYourPetDay	213, 213, 210
#ThingsThatUniteUs	81, 81, 81
#TellTheTruthJoe	62, 61, 61
#VirginiaRally	338, 338, 321

Worldwide	
#GameOfThrones	999, 999, 199
#BoyWithLuv	511, 505, 35
#CES	999, 997, 199
#COVID2019	482, 479, 199
#loveyourpetday	999, 999, 199
#2020NewYear	999, 992, 199
#hanau	367, 349, 190
#InternationalWomensDay	999, 999, 199

Table 5: Amount of tweets for each hashtag (raw tweets resulting from Twitters *English* filter (left), remaining tweets after filtering *English* with *PolyGlot* (middle), and results of manually filtering for spam and non-English tweets after *PolyGlot* (right))

Category	Native	German	Greek	French	Indian	Japanese	Russian	Turkish
Arts/Culture	392	167	165	63	658	174	146	239
Business/Technology/Science	531	116	151	114	125	6	0	136
Politics	625	181	162	74	194	2	0	180
Social/Society	647	102	133	76	158	1	0	243
Total	2195	566	611	327	1135	183	146	798

Table 6: Total amount of tweets for each language and category

As seen in Table 6, we collected similar amounts in each category, with the obvious outliers in Russian and Japanese due to non-existing hashtag data in Business/Technology/Science, Politics and Social/Society.

The datasets were grouped by *Language Family* and separated into even chunks of 100 per group. In case groups had more than one language (Balto-Slavic, Germanic and Romance for Reddit, Indo-European for Twitter) we divided the 100 by the number of languages.

3.1.3. Preprocessing

We apply pre-processing on both datasets to reduce text bloat, clean up any encoding errors (e.g. **& gt ;** is formatted to **<**) and output formatted data which has the same data structure. As can be seen in Figure 3, posts contain character entities[9], redundant spacing and escape characters, which need to be formatted and readable for the text analyzing algorithm. Additionally, platform specific entities such as hashtags, at-mentions and URLs from Twitter media are removed.

We import the Porter corpus of stop words and filter any from the text. We save the filtered text and a separate array of the extracted stop words for later usage. Word elongations and caps-words in the original text are also counted and saved.

Language-check and *aspell* are used to spell-check each word in the filtered text and the first suggested correction is saved in a new array.

3.1.3.1. Levenshtein-distance

We use the *Levenshtein*-distance to calculate the difference between the original text and one that has been checked for spelling mistakes. It is defined by:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } min(i,j) = 0, \\ min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & otherwise. \end{cases} \quad (1)$$

Each result from Language-check and aspell is summed and averaged, and then saved for the classification file.

We calculate text length and the average word length by dividing the length of each word with the text length. The filtered text is then stemmed/lemmatised and tokenised into single words. From these, character tri-grams, word-bigrams and word-unigrams are taken, and a function word uni-gram from the array of function words.

3.1.3.2. Word Stemming Lemmatisation

We lemmatise words to reduce bloat during n-gram creation by converting words which are inflectional or derivative related forms to a common base form.

$$am, are, is \to be$$
$$car, cars, car's, cars' \to car$$

We make use to the *Porter's algorithm*[10] during our work, which 'has repeatedly been shown to be empirically very effective'[11].

[9] https://www.w3.org/TR/REC-html40/sgml/entities.html
[10] https://tartarus.org/martin/PorterStemmer/
[11] https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

Lastly, the stemmed sentences are used for the Parts-of-Speech tagging. For Reddit we use *Tokenizer*[12], which offers token detection for Reddit-specific text. The tokens are then used in the Parts-of-Speech tagger *stanza* (a parser based on Stanford NLP [Qi et al., 2020])[13] and the resulting tags saved in an array. For Twitter we use *TweetNLP*[14] instead, which automatically tokenises and extracts Parts-of-Speech from tweets and is trained on language commonly found on Twitter (e.g. abbreviations and slang). We take Parts-of-Speech bi-grams from the resulting data, count the token occurrence and average it with the text length.

[12] https://github.com/erikavaris/tokenizer
[13] https://github.com/stanfordnlp/stanza/
[14] http://www.cs.cmu.edu/~ark/TweetNLP/

3.2. Feature Extraction

In the following section we introduce the feature selection, and how they were implemented to create the classification dataset for the classes:

1. Native and non-Native English speakers
2. Language family
3. Native language

3.2.1. Features

The features we selected are based on the works of Goldin et al. in order to create a common baseline for evaluation. Following features were also used for our work: *Character tri-grams*, *spelling delta*, *function words* and *sentence length*. Our final feature-set contains 11 different main features and additional sub-features such as Parts-of-Speech tokens and n-gram similarity for the final classification process:

1. **Word elongation**
   Elongated text is commonly found in text messages to emphasise emotional nuance (e.g. "That's sooooo funny"). We use the amount of elongated words to differentiate between more serious categories such as politics, and open categories like social or culture.

2. **Caps usage**
   Similar to elongation, caps is usually used to emphasise emotions like anger or frustration, but can also appear randomly e.g. after unknowingly activating the Caps-Lock function. The amount of words in all-caps were used for this feature.

3. **Text length**
   We use text length to measure text complexity. We assume that, especially due to the character limitation on Twitter, text data from Reddit is on average longer as the website structure benefits long discussions. The total length of the text was used for this feature.

4. **Average Word length**
   Word length is also used to measure text complexity. As Twitter is often used for bursts of short and simple messages we assume that the average word length is lower compared to Reddit. The length of each word was summed and divided by the text length and used for this feature.

5. **Spelling delta**
   We used *language-check*[15] and *aspell*[16] to check text for spelling mistakes and

[15] https://pypi.org/project/language-check/
[16] https://github.com/WojciechMula/aspell-python

replaced any with the first corrected suggestion. The *Levenshtein* distance between the corrected version and the original text from both *aspell* and *language-check* was averaged and used for this feature.

6. **Character tri-grams**
   We extracted all character tri-grams from texts and collected the 1000 most-frequent tri-grams for each class. We calculated n-gram similarity between a given text and each class. The result was used for this feature.

7. **Word bi-grams**
   We extracted all word bi-grams from texts and collected the 300 most-frequent bi-grams for each class. We calculated n-gram similarity between a given text and each class. The result was used for this feature.

8. **Word uni-grams**
   We extracted all word uni-grams from texts and collected the 500 most-frequent uni-grams for each class. We calculated n-gram similarity between a given text and each class. The result was used for this feature.

9. **Function word uni-grams**
   We extracted all function words, or *stop words*, (e.g. he, a, was) from texts and collected the 300 most-frequent uni-grams for each class. We calculated n-gram similarity between a given text and each class. The result was used for this feature.

10. **Parts-of-Speech** (25 tokens)
    We tokenised text into Parts-of-Speech tokens and counted the occurrence. We normalised the count with text length and tokens which are part of Table 7 were used for this feature.

11. **Parts-of-Speech bi-grams**
    We extracted all Parts-of-Speech bi-grams from the tokens generated during the tokenization process and collected the 300 most-frequent bi-grams for each class. We calculated n-gram similarity between a given text and each class. The result was used for this feature.

# Hashtag	@ At-mention	E Emoticon	, Punctuation
& Coordinating conjunction	L Nominal & Verb	Z Proper Noun & possessive	^ Proper Noun
A Adjective	D Determiner	! Interjection	N Common Noun
G Other (e.g. foreign words)	T Verb particle	X Existential there, predeterminers	S Nominal & possessive
R Adverb	U URL or email address	$ Numeral	O Pronoun
~ Discourse marker (e.g. retweet)	V Verb	P Pre- or postposition	Y Predeterminers & verbal
M Proper Noun & verbal			

Table 7: Parts-of-Speech tokens and their definition

3.2.2. N-gram similarity

For the final feature-set we convert n-gram lists to similarity values. First, we import the individual datasets and group the following classes:

1. **Language**
   Reddit: Native, Bulgarian, Croatian, Czech, Lithuanian, Polish, Russian, Serbian, Slovene, German, Finnish, Dutch, Norwegian, Swedish, French, Italian, Portuguese, Romanian, Spanish
   Twitter: Native, German, Greek, French, Indian, Japanese, Russian, Turkish

2. **Language Family**
   Reddit: Native, Romance, Germanic, Balto-Slavic
   Twitter: Native, Indo-European, Turkic, Japonic, Indo-Aryan

3. **Origin**
   Native, non-Native

4. **Category**
   Twitter: Arts/Culture, Business/Technology/Science, Politcs, Social/Society

For each group we count n-gram occurrence and save the 1000 most frequent in descending order. N-grams from each dataset are compared to these and the similarity value to each class is calculated. We define the n-gram similarity as follows:

$$\text{similarity}_{A,B} = \frac{|A \cup B|^n - |A \triangle B|}{|A \cup B|^n} \tag{2}$$

$$A \triangle B = (A \setminus B) \cup (B \setminus A) \tag{3}$$

A and B are two sets of character or word n-grams. We calculate the length of the union of A and B and subtract the length of their symmetric difference. The result is divided by the length of the union of A and B. Sets which have a higher quantity of same elements also have a higher similarity score. n is a *warp* parameter which increases the similarity of shorter strings if $n > 1$.

3.2.3. Feature categories

We separate our features into the two categories i) **Importance-Features** and ii) **TF-IDF Features**.

3.2.3.1. Importance Features

We calculate the importance of features by using the complete datasets as inputs for a *Random Forest* classifier. This process was done on Reddit (European and non-European) and Twitter dataset. We remove features that have zero, or close to zero importance in the over-all classification process.

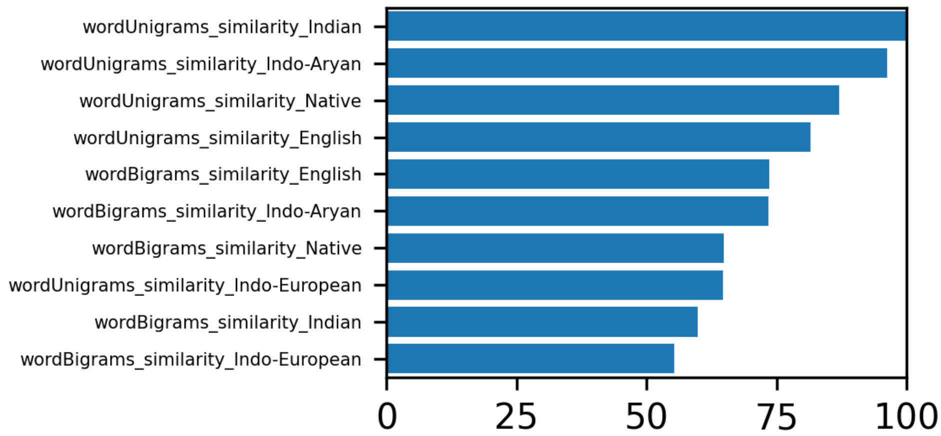

Figure 4: Highest Twitter feature importance scores for **Language Family** from *Random Forest* classifier. Values are normalised to the highest score.

As seen in Table 4, the highest scoring feature was Indo-Aryan word uni-grams followed by Indian word uni-grams. While they are both the same feature (as the Indo-Aryan language family only contains Indian in our dataset), it shows the difference in comparing on a language-level versus language-family. The next three highest are Native word uni-grams, English word uni-grams and English bi-grams. Both language-families are not surprising to see at the top as their relative quantity is higher compared to the other languages. We removed features with zero importance value for the Twitter **Importance-Features** (see Table 22) which are **@**

(At-mention), **Y** (Predeterminers & verbal), **S** (Nominal & possessive), **Z** (Proper Noun & possessive) and **M** (Proper Noun & verbal)

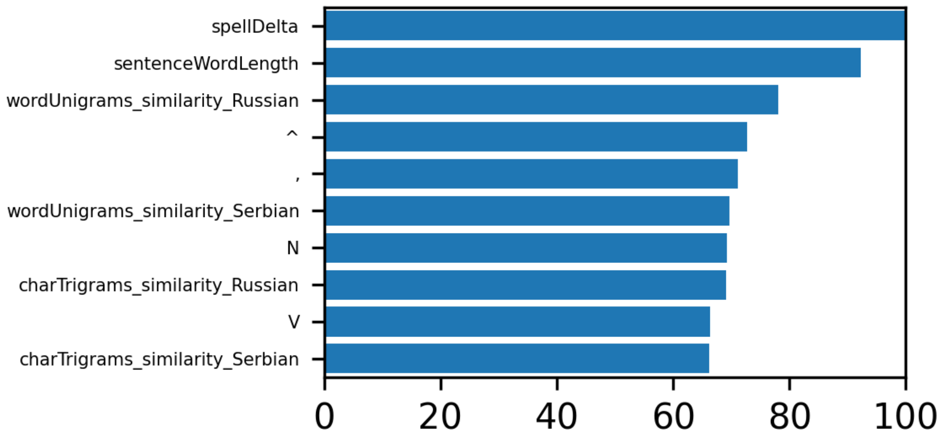

Figure 5: Highest Reddit feature importance scores for **Language** from *Random Forest* classifier in European dataset. Values are normalised to the highest score.

The European Reddit dataset shows spelling delta as its highest scoring feature, followed by average word length. This could be related as longer words are often times more complicated and thus have an increased chance of spelling mistakes. The next highest feature is Russian word uni-grams, followed by proper nouns and punctuation. Balto-Slavic countries seem to represent the top more than others, which implies that they tend to use more unique words. We removed features with zero importance value for the European Reddit **Importance-Features** (see Table 24 which are @ (At-mention), **Y** (Predeterminers & verbal), **S** (Nominal & possessive), **M** (Proper Noun & verbal), **Z** (Proper Noun & possessive), **L** (Nominal & Verb), **X** (Predeterminers), **U** (URL), (Discourse marker) and **E** (Emoticons). Note that most of these features are specific to the Twitter tokeniser (e.g. discourse marker and emoticons) and were not extracted from the Reddit datasets.

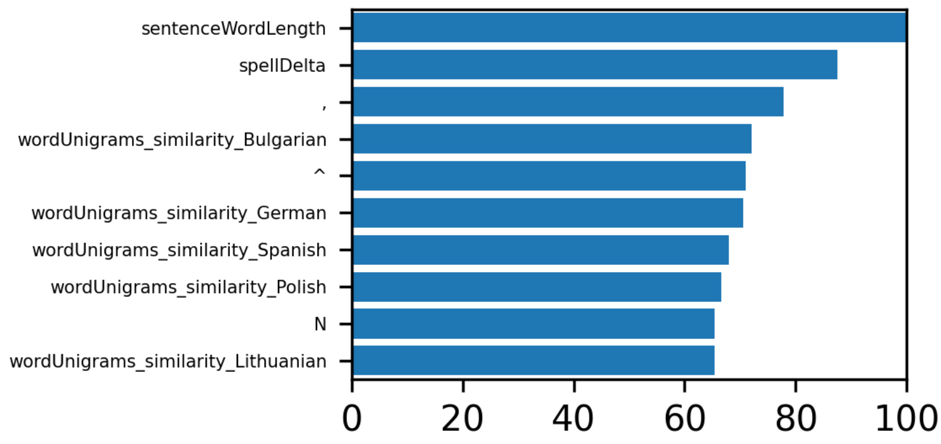

Figure 6: Highest Reddit feature importance scores for **Language** from *Random Forest* classifier in non-European dataset. Values are normalised to the highest score.

The non-European Reddit dataset shows a reversed order for the highest scoring features. Word length is the most important, followed by spelling delta, punctuation, Bulgarian word uni-grams and Proper Nouns. Other than Poland, no Balto-Slavic countries are represented in the top 10, opposite to the European dataset. It seems Russians and Serbians frequent European sub-reddits more commonly and also interact more in those. The switch in the top two could indicate that words in non-European sub-reddits are shorter and thus less likely to contain spelling mistakes. We removed features with zero importance value for the European Reddit **Importance-Features** (see Table 26 which are the same as in the European dataset: **@** (At-mention), **Y** (Predeterminers & verbal), **S** (Nominal & possessive), **M** (Proper Noun & verbal), **Z** (Proper Noun & possessive), **L** (Nominal & Verb), **X** (Predeterminers), **U** (URL), (Discourse marker) and **E** (Emoticons).

3.2.3.2. TF-IDF Features

Term frequency-inverse document frequency or **TF-IDF** is a statistic to represent the importance of a word to a document or corpus. The value increases proportionally to the frequency of the word in a document and is offset by the number of documents it appears in. For this thesis it can help in finding words which are uniquely or more commonly used in certain native languages or language-families. The term frequency is calculated as the raw number of term occurrences in a text. Inverse document frequency is calculated as:

$$idf(t, D) = log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \tag{4}$$

$\boldsymbol{N}$ is the total number of documents in a corpus $\boldsymbol{D}$, the number of documents in which term t appears is $|\boldsymbol{\{d \in D : t \in d\}}|$. TF-IDF is defined as

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D) \tag{5}$$

We convert the lemmatised text into a TF-IDF matrix with *sklearn*[17] TF-IDF Vectorizer and use it as a replacement for n-gram similarity, as we configure it to create uni-, bi- and tri-grams from the text during the process. We compare our definition of n-gram similarity to the performance of term relevancy in the datasets.

3.2.4. Models

We compare an *AutoML* pipeline to basic models such as *Random Forest*, *Logistic regression* and a *Support Vector Machine* in order to test its robustness. We implement **Random Forest** with the criterion *Gini Impurity*, which measures how often a random element would be labelled incorrectly if it was randomly labelled based on the label distribution during the classification process. We also kept the number of trees to the default 100.

The **Support Vector Machine** pipeline consists of a *Standard Scaler* to remove the mean and scale the features to unit variance and a *Linear Support Vector* for classifying the scaler data.

Logistic Regression is initialised with default parameters set by the *sklearn* library. Each classification model is set with a random state of 42 and fittings are accompanied by a 5-fold cross validation.

We create a TPOT classifier with three generations of optimization and a population size of 50. We also make use of 5-fold cross validation during the optimization process. The input is split into chunks of 100 for each language family and then a training and testing set with a ratio of 70:30. We fit the TPOT classifier with the training data and calculate an accuracy score with the test set. A score is taken for each chunk and continued until there have been three chunks without an improvement in accuracy score. The resulting pipeline is exported into a python file and used for the classification process.

The pipeline created by TPOT for Twitter consists of i,ii) *Random Forest* and *Extra Trees* classifier used as stacking estimators which generate predictions used for iii) *Gaussian Naive-Bayes* as the final classification step. For Reddit, TPOT created a pipeline containing a *Linear Support Vector Machine* with hyperparameter optimization. The regularization parameter **C** set to 10, dual-optimization problem as false and a l1 penalty, which creates sparse coefficient vectors as the norm.

[17]https://scikit-learn.org/stable/modules/feature_extraction.html

4. Experiments

In the following section we will first discuss our goal for the experiments by setting a baseline for the results. We explain utilization of the classification files for the experiments and introduce the *TPOT* setup for finding classification pipelines to evaluate the feature performance and compare it to our baseline.

4.1. Tasks

We evaluated three tasks for our experiments based on the work of Volkova et al. and Goldin et al.: i) distinguish between Native and non-Native authors, ii) determine the *Language Family* of non-Native authors, iii) identify the native language of non-Native authors. Additionally, we compare the results from the *TPOT* pipelines to basic classifiers. As we use the same Reddit dataset, we set performance baselines based on their results for these tasks. Goldin et al. managed an average prediction accuracy for feature based classification of 90.77% in-domain (European sub-reddits) and 82.21% out-of-domain (non-European sub-reddits) in binary classification (Native and non-Native), 78.31% and 57.90% in language family, and 63.04% and 32.73% in native language identification. From these results we set our baseline as follows:

Dataset	Origin i)	Language Family ii)	Language iii)
European	90.77%	78.31%	63.04%
non-European	82.21%	57.9%	32.73%

Table 8: Baseline prediction accuracy for each research task and dataset

As the Twitter dataset was newly created we used a trivial baseline for binary classification tasks of 50% prediction accuracy for *Origin* and *Category*, 20% for *Language Family* and 12.5% for *Language*.

From these tasks and related works we derived three research questions we answer in this thesis: i) How strongly does text style differ cross-platform and among different domains? ii) Can native language identification from English text solely based on linguistic features obtain accurate results? iii) How does an automated machine learning pipeline perform compared to basic classification models on language identification tasks?

4.2. Methods

First, we extract the label data from the dataset, encode it with a *Label Encoder* and initialise the classifiers and TPOT pipelines. We create the chunks with the **Importance-Features** for the given dataset and split each chunk into 70% training and 30% testing sets. The training sets are fitted to each of the classification models, followed by a prediction on the testing sets. We save the prediction output and the actual labels for each class, classification model and dataset. Classification models are re-initialised for each class. Next we initialize a *TF-IDF Vectorizer* with unigrams, bi-grams, and tri-grams and a maximum feature amount of 2000 for Twitter, and 1500 for Reddit due to limited computational power. We fit the Vectorizer with lemmatised text from the dataset and generate a feature matrix, which we append to the general feature-set (every feature except n-gram similarity). Chunks are split into 70% training and 30% testing sets again, used to fit the classification models and predict the classes. Lastly, we calculate *Accuracy*, *F1 macro* and *Precision* for each prediction we generated and save it. We define accuracy as

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{6}$$

where *tp* are the true positive predictions, *tn* true negatives, *fp* false positives and *fn* false negatives. Precision is defined as

$$\text{Precision} = \frac{tp}{tp + fn} \tag{7}$$

and F1 macro as

$$F_2 = \frac{5 \cdot tp}{5 \cdot tp + 4 \cdot fn + fp} \tag{8}$$

4.3. Results

We separate our results in the three datasets and the subsets *Language*, *Language Family* and *Origin*. For Twitter we include an additional subset for *Category*. We present scores for mean prediction accuracy, F1 macro and precision for *Random Forest*, *Logistic Regression*, *Support Vector Machine* and the *TPOT* pipelines on the given dataset.

4.3.1. Reddit dataset

The following tables show the results of the European and non-European Reddit dataset used as input for the classification process with either Importance-Features or TF-IDF features enabled.

Features	Model	Class	Accuracy	F1 macro	Precision
Importance	Random Forest	Origin	0.71 ±0.06	0.53 ±0.08	0.66 ±0.08
		Language Family	0.35 ±0.13	0.35 ±0.13	0.37 ±0.13
		Language	0.29 ±0.1	0.09 ±0.03	0.19 ±0.08
	Support Vector Machine	Origin	0.69 ±0.06	0.58 ±0.06	0.67 ±0.06
		Language Family	0.42 ±0.09	0.42 ±0.09	0.44 ±0.09
		Language	0.27 ±0.08	0.1 ±0.04	0.2 ±0.09
	Logistic Regression	Origin	0.7 ±0.07	0.53 ±0.07	0.65 ±0.08
		Language Family	0.35 ±0.12	0.34 ±0.12	0.37 ±0.12
		Language	0.24 ±0.08	0.1 ±0.04	0.2 ±0.09
	AutoML Model	Origin	0.71 ±0.06	0.55 ±0.08	0.67 ±0.07
		Language Family	0.39 ±0.11	0.39 ±0.11	0.42 ±0.11
		Language	0.26 ±0.08	0.1 ±0.05	0.2 ±0.09
TF-IDF	Random Forest	Origin	0.72 ±0.05	0.5 ±0.08	0.67 ±0.11
		Language Family	0.39 ±0.12	0.38 ±0.12	0.43 ±0.12
		Language	0.32 ±0.09	0.09 ±0.03	0.19 ±0.07
	Support Vector Machine	Origin	0.65 ±0.06	0.59 ±0.06	0.67 ±0.06
		Language Family	0.39 ±0.12	0.38 ±0.12	0.4 ±0.11
		Language	0.21 ±0.07	0.13 ±0.04	0.24 ±0.08
	Logistic Regression	Origin	0.7 ±0.07	0.55 ±0.07	0.66 ±0.07
		Language Family	0.35 ±0.12	0.34 ±0.12	0.37 ±0.12
		Language	0.24 ±0.08	0.11 ±0.04	0.2 ±0.09
	AutoML Model	Origin	0.71 ±0.06	0.72 ±0.07	0.69 ±0.07
		Language Family	0.4 ±0.11	0.4 ±0.11	0.42 ±0.1
		Language	0.25 ±0.09	0.15 ±0.04	0.25 ±0.08

Table 9: Results of classification for the European Reddit dataset. The highest values in each class and score are highlighted in grey.

Features	Model	Class	Accuracy	F1 macro	Precision
Importance	Random Forest	Origin	0.73 ±0.06	0.58 ±0.09	0.70 ±0.07
		Language Family	0.40 ±0.12	0.40 ±0.13	0.42 ±0.12
		Language	0.31 ±0.11	0.12 ±0.05	0.23 ±0.10
	Support Vector Machine	Origin	0.73 ±0.07	0.64 ±0.09	0.71 ±0.08
		Language Family	0.40 ±0.13	0.39 ±0.13	0.42 ±0.13
		Language	0.29 ±0.09	0.11 ±0.06	0.21 ±0.10
	Logistic Regression	Origin	0.72 ±0.05	0.57 ±0.09	0.68 ±0.07
		Language Family	0.40 ±0.13	0.39 ±0.13	0.42 ±0.13
		Language	0.26 ±0.09	0.12 ±0.05	0.23 ±0.11
	AutoML Model	Origin	0.74 ±0.06	0.61 ±0.10	0.71 ±0.08
		Language Family	0.40 ±0.13	0.39 ±0.14	0.42 ±0.13
		Language	0.27 ±0.09	0.11 ±0.05	0.21 ±0.10
TF-IDF	Random Forest	Origin	0.74 ±0.05	0.55 ±0.10	0.72 ±0.11
		Language Family	0.44 ±0.13	0.43 ±0.13	0.47 ±0.12
		Language	0.34 ±0.09	0.11 ±0.04	0.23 ±0.08
	Support Vector Machine	Origin	0.68 ±0.07	0.63 ±0.07	0.71 ±0.06
		Language Family	0.44 ±0.11	0.43 ±0.12	0.45 ±0.11
		Language	0.26 ±0.07	0.17 ±0.05	0.30 ±0.10
	Logistic Regression	Origin	0.73 ±0.05	0.58 ±0.09	0.69 ±0.07
		Language Family	0.40 ±0.13	0.40 ±0.13	0.42 ±0.13
		Language	0.25 ±0.09	0.13 ±0.05	0.23 ±0.11
	AutoML Model	Origin	0.73 ±0.06	0.64 ±0.07	0.72 ±0.06
		Language Family	0.44 ±0.12	0.43 ±0.12	0.46 ±0.11
		Language	0.30 ±0.09	0.18 ±0.05	0.30 ±0.09

Table 10: Results of classification for the non-European Reddit dataset. Highest values in each class and score are highlighted in grey.

4.3.2. Twitter dataset

The following Table shows the results of Twitter dataset used as input for the classification process with either Importance-Features or TF-IDF features enabled.

Features	Model	Class	Accuracy	F1 macro	Precision
Importance	Random Forest	Origin	0.94 ±0.08	0.91 ±0.12	0.94 ±0.08
		Language Family	0.8 ±0.23	0.8 ±0.24	0.8 ±0.23
		Language	0.56 ±0.12	0.47 ±0.18	0.55 ±0.15
		Category	0.8 ±0.18	0.69 ±0.24	0.79 ±0.19
	Support Vector Machine	Origin	0.88 ±0.05	0.84 ±0.06	0.89 ±0.05
		Language Family	0.46 ±0.06	0.45 ±0.06	0.46 ±0.07
		Language	0.5 ±0.17	0.44 ±0.21	0.49 ±0.19
		Category	0.72 ±0.17	0.58 ±0.22	0.74 ±0.19
	Logistic Regression	Origin	0.78 ±0.04	0.63 ±0.09	0.75 ±0.05
		Language Family	0.49 ±0.06	0.48 ±0.08	0.51 ±0.06
		Language	0.58 ±0.15	0.49 ±0.21	0.56 ±0.17
		Category	0.73 ±0.18	0.59 ±0.22	0.75 ±0.2
	AutoML Model	Origin	0.88 ±0.06	0.82 ±0.07	0.87 ±0.06
		Language Family	0.47 ±0.06	0.45 ±0.06	0.46 ±0.06
		Language	0.52 ±0.2	0.45 ±0.24	0.5 ±0.21
		Category	0.74 ±0.18	0.59 ±0.23	0.75 ±0.19
TF-IDF	Random Forest	Origin	0.94 ±0.08	0.89 ±0.14	0.94 ±0.07
		Language Family	0.85 ±0.18	0.84 ±0.19	0.86 ±0.17
		Language	0.65 ±0.1	0.55 ±0.17	0.65 ±0.15
		Category	0.82 ±0.15	0.7 ±0.22	0.83 ±0.15
	Support Vector Machine	Origin	0.83 ±0.06	0.79 ±0.07	0.85 ±0.05
		Language Family	0.63 ±0.05	0.63 ±0.07	0.65 ±0.06
		Language	0.66 ±0.13	0.61 ±0.16	0.67 ±0.13
		Category	0.75 ±0.16	0.6 ±0.16	0.83 ±0.13
	Logistic Regression	Origin	0.83 ±0.04	0.73 ±0.07	0.83 ±0.05
		Language Family	0.64 ±0.07	0.64 ±0.08	0.66 ±0.06
		Language	0.64 ±0.12	0.55 ±0.19	0.62 ±0.14
		Category	0.77 ±0.17	0.64 ±0.22	0.79 ±0.18
	AutoML Model	Origin	0.86 ±0.05	0.79 ±0.06	0.85 ±0.05
		Language Family	0.63 ±0.07	0.63 ±0.08	0.65 ±0.06
		Language	0.65 ±0.13	0.6 ±0.16	0.67 ±0.14
		Category	0.78 ±0.15	0.65 ±0.21	0.81 ±0.16

Table 11: Results of classification for the Twitter dataset. The highest values in each class and score are highlighted in grey.

5. Evaluation

In the following section we will evaluate and discuss the results of the classification. We split the discussion into four parts.

i) Evaluating the prediction results and comparison to the baseline, ii) Comparing the results from the basic classifiers to the *TPOT* pipelines, iii) Discussing the differences between the *Importance-Features* and *TF-IDF*, iv) Investigating the text style in *Language*, *Language Family*, *Origin* and *Category*.

5.1. Classification

5.1.1. Results

Class	Dataset	Baseline	Accuracy	F1 Macro	Precision
Origin	European	90.77%	72% ±5	59% ±7	69% ±7
	Non-European	82.21%	74% ±5	64 ±7	72% ±6
	Twitter	50%	94% ±8	91% ±12	94% ±8
Language Family	European	78.31%	42% ±9	42% ±9	44% ±9
	Non-European	57.9%	44% ±12	43% ±12	47% ±12
	Twitter	20%	85% ±18	84% ±19	86% ±17
Language	European	63.04%	32% ±9	15% ±4	25% ±8
	Non-European	32.73%	34% ±9	18% ±5	30% ±9
	Twitter	12.5%	66% ±13	61% ±16	67% ±13
Category	Twitter	50%	82% ±15	70% ±22	83% ±13

Table 12: Baseline accuracy assumptions compared to best results from our classification. Values that are higher than the baseline are highlighted in grey.

The results for the **Reddit** dataset are lower compared to the baseline as seen in Table 12. *Origin* scores for the European dataset are 18.77% lower at 72% and non-European at 74%. F1 and precision scores are, in most cases, close to the accuracy value. In *Language Family* we observe a similar depiction. The European dataset obtained 42% correct predictions compared to the 78.31% baseline, whereas the non-European scores are 44%, 13.9% lower than the baseline. F1 and precision scores are also within ~3% of accuracy scores. In *Language* we obtained 31.04% lower mean accuracy in the European dataset, and 1.27% higher in the non-European. While precision scores are again within range of accuracy scores, F1 scores are ~50% lower than those of accuracy.

Twitter results for *Origin* are way above our expectations with 94% mean prediction accuracy compared to the 50% baseline. *Language Family* is also higher than our assumptions with the best average at 85% - 65% above the baseline. We see the same trend in *Language* with 66% prediction accuracy compared to 12.5% baseline, and in *Category* with 82% to 50%. F1 and precision scores are close to accuracy scores and within standard deviation range. While the Twitter dataset was created and annotated from scratch we observe values more in line to the baseline set for Reddit, however standard deviation values are also almost double that of Reddit's in most metrics. A balanced dataset which includes more samples for Japanese and Russian might decrease these ranges.

5.1.2. Models

Model	Class	Accuracy	F1 macro	Precision
Random Forest	Origin	94% ±8	91% ±12	94% ±8
	Language Family	85% ±18	84% ±19	86% ±17
	Language	65% ±10	55% ±17	65% ±15
	Category	82% ±15	70% ±22	83% ±15
Support Vector Machine	Origin	88% ±6	84% ±7	89% ±5
	Language Family	63% ±6	63% ±7	65% ±7
	Language	66% ±17	61% ±21	67% ±19
	Category	75% ±17	60% ±22	83% ±19
Logistic Regression	Origin	83% ±4	73% ±9	83% ±5
	Language Family	64% ±7	64% ±8	66% ±6
	Language	64% ±15	55% ±21	62% ±17
	Category	77% ±18	64% ±22	79% ±20
AutoML Model	Origin	88% ±6	82 ±7	87 ±6
	Language Family	63% ±7	63% ±8	65% ±6
	Language	65% ±20	60% ±24	67% ±21
	Category	78% ±18	65% ±23	81% ±19

Table 13: Comparison between classification model scores for the **Twitter** dataset. Results for different features are merged

Random Forest obtained accuracy, F1 and precision scores 5 to 7% higher in *Origin* compared to other models, 20 to 22% higher in *Language Family* and up to 10% higher in *Category* as seen in Table 13. It also shows the highest standard deviation scores in language family classification with 17 to 19% on average. *Support Vector Machine* obtained higher scores in *Language* for accuracy (66%), F1 (61%) and precision (67%). *AutoML* and *Logistic Regression* obtain similar scores in language, language family and category but are lower than Random Forest and Support Vector Machine. In *Origin*, AutoML manages scores similar to the second highest with 88% accuracy, 82% F1 and 87% precision scores.

Model	Class	Accuracy	F1 Macro	Precision
Random Forest	Origin	72% ±5	53% ±8	67% ±11
	Language Family	39% ±12	38% ±12	43% ±12
	Language	32% ±9	9% ±3	19% ±7
Support Vector Machine	Origin	69% ±6	59% ±6	67% ±6
	Language Family	42% ±9	42% ±9	44% ±9
	Language	27% ±8	10% ±4	20% ±9
Logistic Regression	Origin	70% ±7	55% ±7	66% ±7
	Language Family	35% ±12	34% ±12	37% ±12
	Language	24% ±8	11% ±4	20% ±9
AutoML Model	Origin	71% ±6	59% ±7	69% ±7
	Language Family	40% ±11	40% ±11	42% ±10
	Language	26% ±8	15% ±4	25% ±8

Table 14: Comparison between classification model scores for the **European Reddit** dataset. Results for different features are merged

The highest mean accuracy scores in *Origin* and *Language* are obtained by *Random Forest* at 72% and 32% respectively as seen in Table 14. In F1 and precision the *AutoML* pipeline scores higher at 59%, 69% and 15%, 25% respectively. *Support Vector Machine* obtains the highest and most consistent prediction scores in *Language Family* at 42% accuracy, 42% F1 and 44% precision, on average 1~2% higher than the second highest scores. *Logistic Regression* obtains the lowest scores in most metrics in *Language Family* and *Language*. While *AutoML* scored lower in accuracy for *Language*, its F1 and precision are the highest. Scores in other classes are also similar to the highest, or are the highest.

Model	Class	Accuracy	F1 Macro	Precision
Random Forest	Origin	74% ±5	58% ±9	72% ±11
	Language Family	44% ±13	43% ±13	47% ±12
	Language	34% ±9	12% ±5	23% ±8
Support Vector Machine	Origin	73% ±7	64% ±9	71% ±6
	Language Family	44% ±11	43% ±12	45% ±11
	Language	29% ±9	17% ±5	30% ±10
Logistic Regression	Origin	73% ±5	58% ±9	69% ±7
	Language Family	40% ±13	40% ±13	42% ±13
	Language	26% ±9	13% ±5	23% ±11
AutoML Model	Origin	74% ±6	64% ±7	72% ±6
	Language Family	44% ±12	43% ±12	46% ±11
	Language	30% ±9	18% ±5	30% ±9

Table 15: Comparison between classification model scores for the **non-European Reddit** dataset. Results for different features are merged.

AutoML obtained on average the highest scores in *Origin* at 74% accuracy, 64% F1 and 72% precision; Other classification models scored similarly as seen in Table 15. In *Language Family* TPOT and *Random Forest* are within 1% range of each other, with the highest scores at 44% accuracy, 43% F1 and 47% precision. Random Forest obtained the highest accuracy score in *Language* at 34% and 32%, while AutoML scored highest in F1 and precision at 18% and 30% respectively. *Logistic Regression* obtained the lowest scores on average in all classes.

We observe that in most cases *AutoML* was able to obtain the highest accuracy scores or was within ~5% of other models. Thus, we can assume that an automated machine learning pipeline such as AutoML can obtain results that are on par with traditional classification models for our proposed task.

5.1.3. Features

Features	Class	Accuracy	F1 Macro	Precision
Importance	Origin	71% ±6	58% ±6	67% ±6
	Language Family	42% ±9	42% ±9	44% ±9
	Language	29% ±10	10% ±4	20% ±9
TF-IDF	Origin	72% ±5	59% ±7	69% ±7
	Language Family	40% ±11	40% ±11	43% ±12
	Language	32% ±9	15% ±4	25% ±8

Table 16: Comparison of mean scores between *Importance* and *TF-IDF* features for each class in the **European Reddit** dataset. Results for different models are merged.

TF-IDF obtains 1~3% higher prediction scores in both *Origin* and *Language* in Table 16. The highest difference is F1 and precision score in language; TF-IDF obtained 15% in F1 compared to 10%, and 25% precision to 20% in *Importance*. In *Language Family* Importance features obtain higher and more consistent results in each score, 42% in accuracy, 42% in F1 and 44% in precision.

Features	Class	Accuracy	F1 Macro	Precision
Importance	Origin	74% ±6	64% ±9	71% ±8
	Language Family	40% ±12	40% ±13	42% ±12
	Language	31% ±11	12% ±5	23% ±10
TF-IDF	Origin	74% ±5	64% ±7	72% ±6
	Language Family	44% ±13	43% ±12	47% ±12
	Language	34% ±9	18% ±5	30% ±9

Table 17: Comparison of mean scores between *Importance* and *TF-IDF* features for each class in the **non-European Reddit** dataset. Results for different models are merged.

TF-IDF obtains the highest scores in all classes in Table 17, with the highest difference in *Language Family* and *Language* at 3~5% and 2~8% higher scores on average. While *Importance* features are within 1~2% in *Origin*, they are also less consistent compared to TF-IDF.

Features	Class	Accuracy	F1 macro	Precision
Importance	Origin	94% ±8	91% ±12	94% ±8
	Language Family	80% ±23	80% ±24	80% ±23
	Language	58% ±20	49% ±24	56% ±21
	Category	80% ±18	69% ±24	79% ±20
TF-IDF	Origin	94% ±8	89% ±14	94% ±7
	Language Family	85% ±18	84% ±19	86% ±17
	Language	66% ±13	61% ±19	67% ±15
	Category	82% ±17	70% ±22	83% ±18

Table 18: Comparison of mean scores between *Importance* and *TF-IDF* features for each class in the **Twitter** dataset. Results for different models are merged.

We observe the highest score delta in the *Twitter* dataset as seen in Table 18. *TF-IDF* increases prediction scores by 4~6% in *Language Family* compared to *Importance* features and is more consistent (5 to 6% lower standard deviation in all scores). In *Language* scores increase by 8% in accuracy up to 11% in precision and 12% in F1. Accuracy in *Category* increases by 2% in TF-IDF, F1 by 1% and precision by 4%. Importance features obtain similar scores in *Origin* however: 94% accuracy and 94% precision, and increases the F1 score by 2%. We assume the features which were created by the TF-IDF Vectorizer during the n-gram tokenisation establish more distinctive language profiles. Except for *Origin*, classes have a higher standard deviation compared to the results from Reddit. Most are in the range of 17 to 23%, compared to an average increase of 8 to 12% in Reddit. We assume this is due to the unbalanced dataset and our chunking process, creating chunks which do not contain all languages.

To represent the prediction accuracy scores we create a confusion matrix by comparing the true label to the predicted label. Darker cells show a higher quantity of entries classified as that specific class. A perfect prediction would show up as a diagonal line from top left to bottom right.

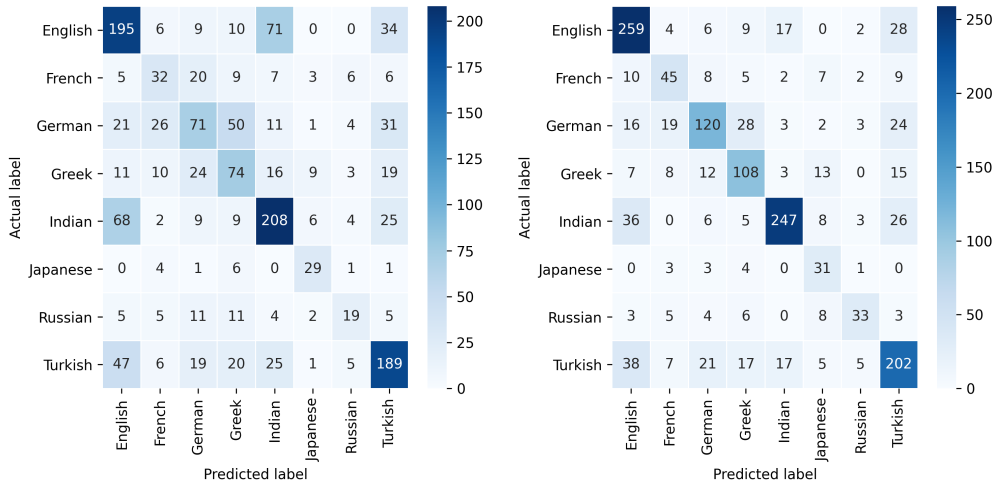

Figure 7: Confusion matrix for *Language* in the Twitter dataset between **Importance** (left) and **TF-IDF** (right)

True positives in the class *English* are higher for TF-IDF as seen in Figure 7, with most of the difference stemming from *Indian* (In *Importance* 71 native English texts were falsely classified as Indian, in *TF-IDF* just 17). Another noticeable change is the hot-spot with German and Greek, which is more clear and defined in TF-IDF. In general it seems that the classifier can identify Indian text better with TF-IDF, which also increases prediction accuracy for other languages that were falsely classified as Indian.

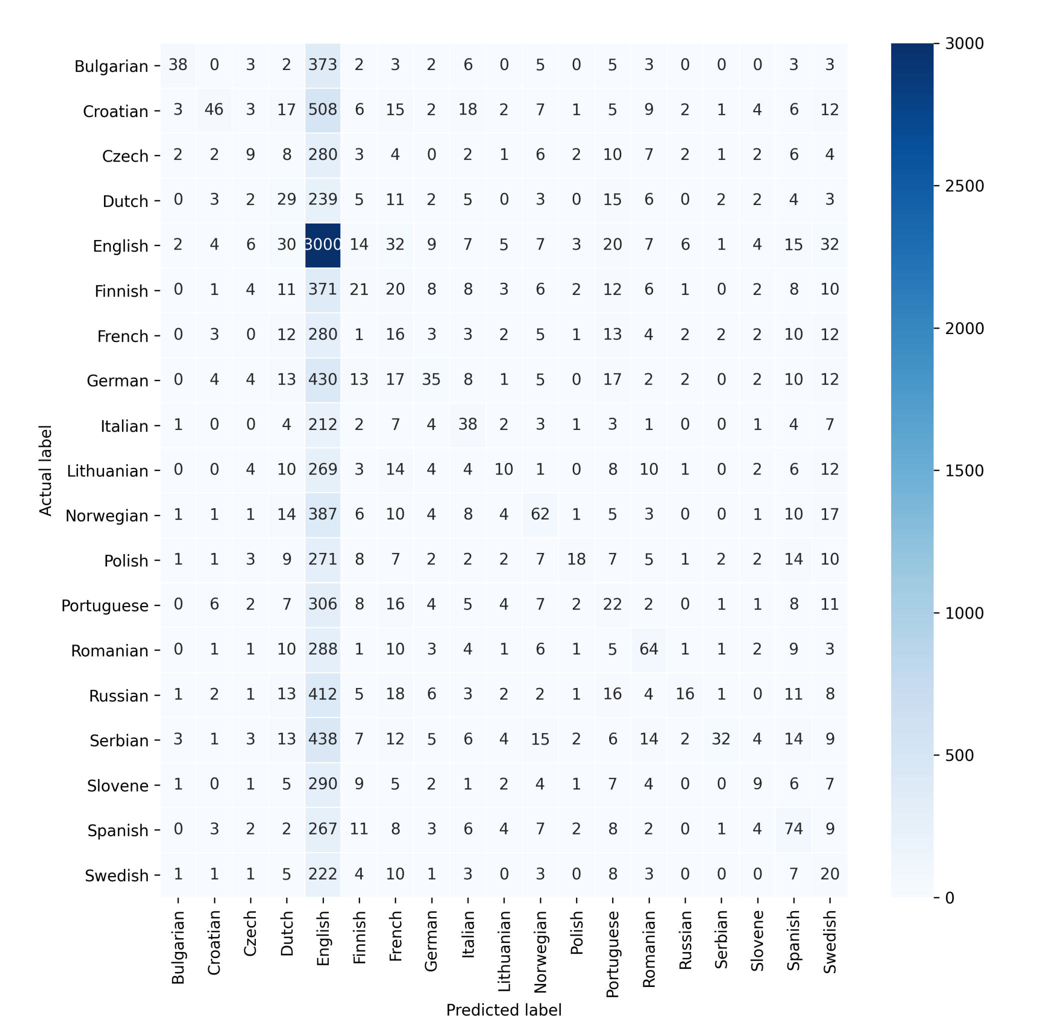

Figure 8: Confusion matrix for *Language* in the non-European Reddit dataset

In comparison, the confusion matrix for *Language* in Figure 8 shows a clear lack of distinction between English and other languages. The only class that was predicted mostly correctly is English (93.6% true positives), with a trend of 70-80% of other text also being classified as English. As the datasets were very balanced with equal amounts of text for each language, we can only assume that text on Reddit is a lot more similar than Twitter. Our hypothesis is that most users on Reddit tend to check what they are writing (e.g. spelling or grammar check) before committing, thus decreasing the over-all mistake rate, which we assume to be higher due to longer texts on average.

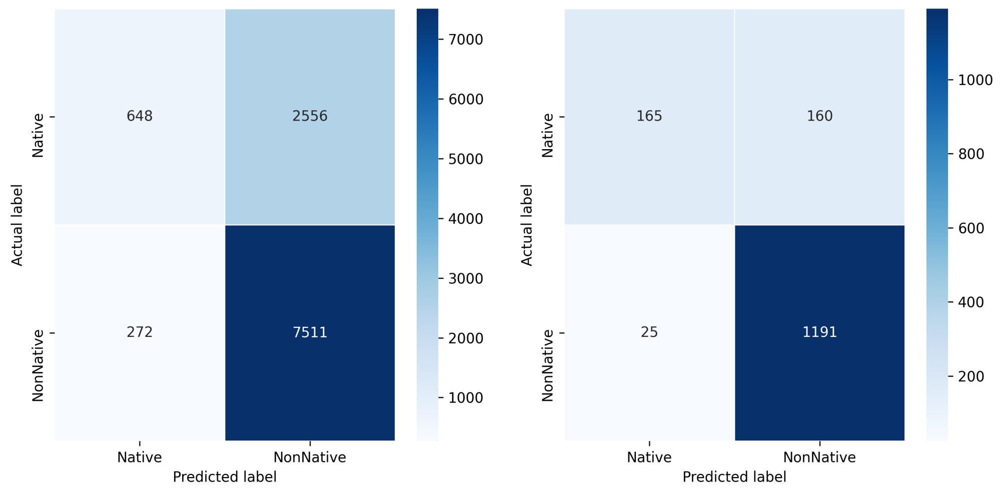

Figure 9: Confusion matrix for *Origin* in the non-European Reddit (left) and Twitter (right) dataset

We observe a 75:25 split of true positives and false negatives for *Origin* in Figure 9, but an overwhelming 97.4% true negative rate. We assume the occurrence due to the unbalanced amount of Native to non-Native languages, as we can clearly identify the same trend in the Twitter dataset. The differentiation between Native and non-Native seems stronger on Twitter however, as their ratio of true positives to false negatives is only ~50%.

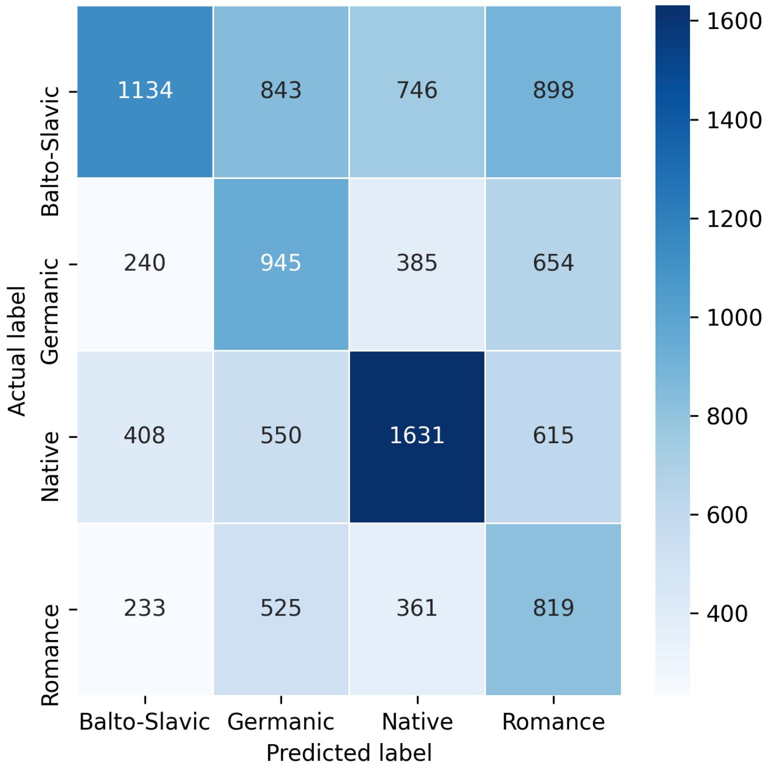

Figure 10: Confusion matrix for *Language Family* in the non-European Reddit dataset

The prediction rate of Native users is the highest in the non-European Reddit dataset (49.1%) seen in Figure 10. The Germanic family obtains over 42.4% correct predictions and Romance over 42.2%. Balto-Slavic has the lowest prediction score with 31.3%, showing that this language family is harder to distinguish from others. We assume that a higher amount of highly fluent Balto-Slavic users are posting on Reddit compared to other language families.

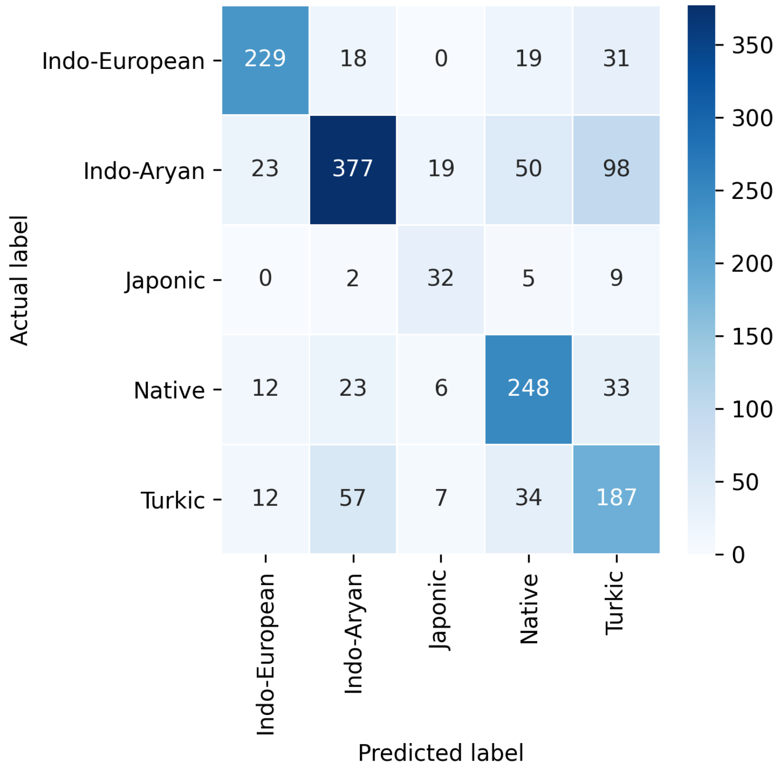

Figure 11: Confusion matrix for *Language Family* in the Twitter dataset

The highest factor for false classification is Indo-Aryan with 33.5% wrong predictions (as seen in Figure 11) since it is also the largest group. Indo-European, Japonic and Native contain high true positive rates with 77.1%, 66.7% and 77% respectively. Turkic obtained 62.9% correct predictions, which is similar to Indo-Aryan but also has less impact over-all as it is almost half the size in samples. We observe that these two families share the highest similarity as 98 Indo-Aryan samples were classified as Turkic, and 57 Turkic as Indo-Aryan.

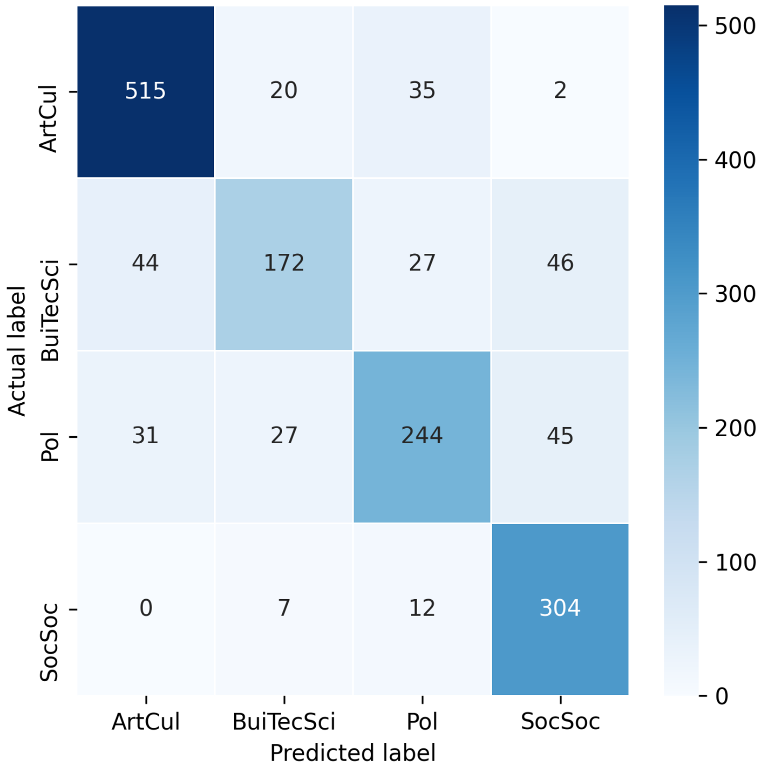

Figure 12: Confusion matrix for *Category* in the Twitter dataset

Arts/Culture and Social/Society have the highest true positive rates at 90% and 94% respectively in Figure 12. These categories seem to have a clear distinction in language style compared to the other categories we label as 'serious' such as Politics. Politics and Business/Technology/Science seem to be less defined. We assume this is due to the nature of these 'serious' topics which are most likely longer texts and words, and contain less adjectives as they do not favour emotional and expressional language. The following sample text was classified as Business/Technology/Science, while its actual class is Politics:

> Why does Joe Biden's campaign keep going after victims of the opioid epidemic while taking money from big pharma? Seems corrupt to me.

We assume words such as 'money' and 'epidemic' appear more often in Business and Science which enabled this wrong prediction.

5.2. Language

We also investigated differences in language style between Languages, Language Families, Categories and Origins.

5.2.1. Twitter

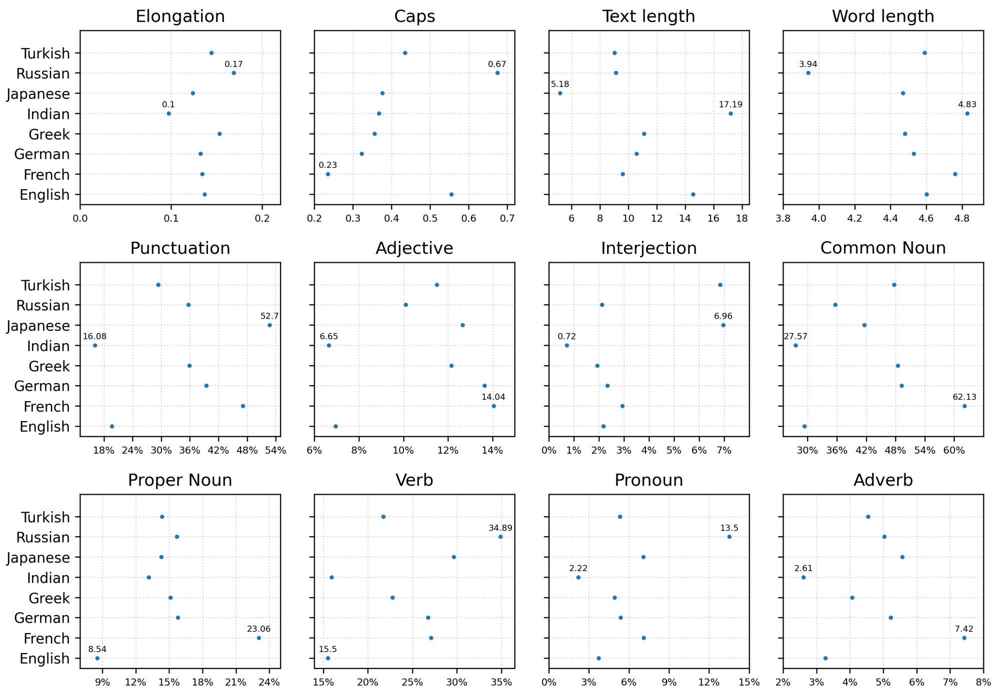

Figure 13: Average values of common language traits in Twitter dataset grouped by **Language**. Elongation, caps, text length and word length are average quantity values. Other values show average percentage of text which is that specific trait.

Russian users have an increased usage of elongation (average 0.168 per text) and caps (0.67) in their text as seen in Figure 13. They are also the most prominent users of pronouns (13.5%). Indians have both the longest texts (on average 17.2 words) and words (4.82 characters per word). Japanese use considerably more interjections (6.95%) and punctuation (52.69%), however we assume this is due to the low data variety.

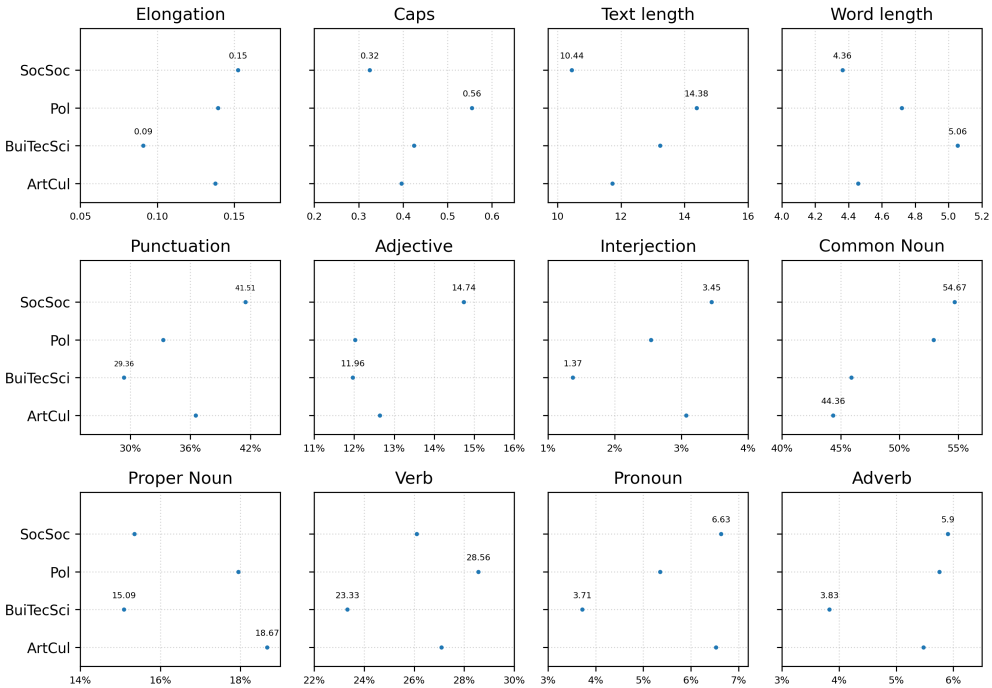

Figure 14: Average values of common language traits in Twitter dataset grouped by **Category**. Elongation, caps, text length and word length are average quantity values. Other values show average percentage of text which is that specific trait.

We observe some obvious trends in language usage in Figure 14. Business/ Technology/Science contains the lowest amount of elongations (average 0.091 per text), but also the longest words (5.06 characters per word). Text in this category has the lowest percentage of any Parts-of-Speech compared to the others. Social/Society has the shortest text (10.44 words) and word (4.36 characters) length and contains the least amount of caps (0.32) . It also contains the highest amount of elongations (0.152) and the highest punctuation (41.5%) and adjective (14.74%) usage. Text in Politics has the highest amount of caps (0.55) and is the longest (14.38 words). It also contains the highest amount of verbs (28.56%).

	Character tri-grams	Word bi-grams	Word uni-grams	Parts-of-Speech bi-grams	Function words
	the, 231	it s, 46	amp, 116	N N, 1420	the, 720
	day, 199	small busi, 41	gun, 97	A N, 787	to, 523
Native	all, 191	busi confer, 27	today, 86	V N, 711	a, 368
	con, 183	i m, 19	us, 81	N V, 561	and, 358
	rea, 176	nd amend, 19	day, 78	V V, 441	of, 355
	tur, 180	ann ann, 49	turkish, 66	N N, 849	the, 432
	urk, 141	turkish armi, 16	day, 60	A N, 418	to, 283
Turkic	the, 134	allah give, 12	turkey, 58	V N, 373	of, 219
	day, 127	may allah, 11	ann, 50	N V, 293	in, 204
	rea, 112	it s,11	world, 45	^^, 270	and, 195
	day, 230	it s , 25	we, 94	N N, 1427	the, 790
	rea, 226	i want, 22	today, 90	A N, 808	to, 567
Indo-European	man, 204	german armi, 20	day, 90	V N, 683	in, 417
	the, 203	let s, 18	greec, 86	N V, 570	of, 386
	gre, 202	i am, 15	it, 71	^^, 452	and, 366
	ove, 16	k rt, 7	k, 14	N N, 35	you, 23
	aaa, 15	rt k, 7	good, 10	A N, 29	to, 17
Japonic	you, 13	k follow, 7	love, 10	V V, 25	the, 13
	asu, 13	follow k, 6	follow, 10	V N, 25	is, 12
	www, 13	the world, 5	you, 9	N V, 22	a, 11
	amp, 430	rampal ji, 278	ji, 312	N N, 1510	the, 900
	har, 356	ji maharaj, 230	rampal, 282	^^, 1052	of, 659
Indo-Aryan	ram, 342	saint rampal, 209	maharaj, 249	A N, 911	is, 496
	int, 327	golden age, 71	saint, 235	N V, 714	to, 483
	ain, 316	must watch, 63	come, 201	V N, 691	in, 435
	the, 855	rampal ji, 279	ji, 313	N N, 5250	the, 2860
	rea, 733	ji maharaj, 231	come, 308	A N, 2966	to, 1881
Total	day, 672	saint rampal, 210	amp, 307	V N, 2489	of, 1629
	amp, 659	it s, 96	rampala, 283	^^, 2199	in, 1357
	ter, 648	golden age, 71	today, 269	N V, 2159	and, 1324

Table 19: Highest occurring n-gram values and their quantity in each language family from Twitter.

Character tri-grams and word uni-grams in Table 19 show a trend of including country names, which could help immensely in identifying the native language. Parts-of-Speech are similar, with obvious outliers proper noun frequency in the Indo-Aryan family. Function words are similar both in ranking and entries, only Native speakers make more use of *a* compared to others. Word bi-grams are more varied and contain hints to the source hashtags that were used to create the datasets (e.g. *saint rampal*, *rampal ji*).

5.2.2. Reddit

We compared European and non-European data in *Language*, *Language Family* and *Origin* classes.

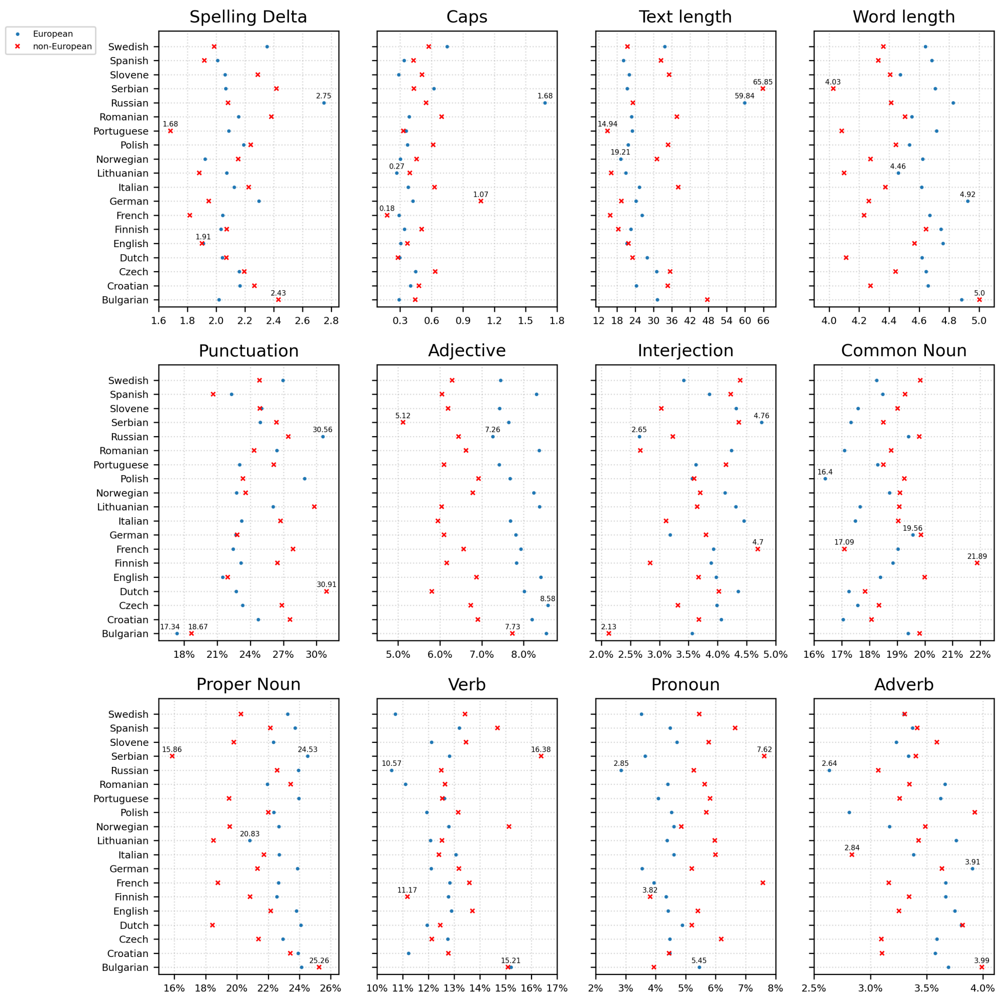

Figure 15: Average values of common language traits in Reddit dataset grouped by **Language**. Spelling delta, caps, text length and word length are average quantity values. Other values show average percentage of text which is that specific trait.

Observing the European data in Figure 15, the most obvious outlier is Russian. It far outscores other languages in the European dataset with a spelling delta of

2.747 (versus second highest Swedish with 2.35), average caps of 1.68 words per text (to 0.74 in Swedish) and a text length of **59.84** words, with 33.6 in comparison from Swedish. Average Russian text length is almost double that of the second highest, which also explains why their spelling delta is higher than others. Other notable features for Russian are the highest percentage of punctuation (30.56%) and the lowest percentage of adjectives (7.26%), interjections (2.65%), verbs (10.56%), pronouns (2.85%) and adverbs (2.63%). Non-European data shows a different picture; Portuguese (1.68), French (1.81) and Lithuanian (1.88) have a lower spelling delta than Native speakers, which scored almost similar in both European (1.91) and non-European (1.90) sub-reddits. Serbian average text length is higher than Russians' with **65.85** words, while simultaneously having the shortest words (4.03 characters). Contrary to Russian in the European dataset, Serbian text contains the highest percentage of verbs (16.38%) and pronouns (7.62%), but also the lowest percentage of proper nouns (15.86%), which is in stark contrast to the European dataset where Serbian had the highest. Adjectives also show an interesting trend: Except for the highest percentage in Bulgarian (7.73) in the non-European dataset, every other language scores consistently higher, with the lowest being Russian (7.26). Similar trends can be seen in word length, proper nouns and reversed for pronouns and common nouns.

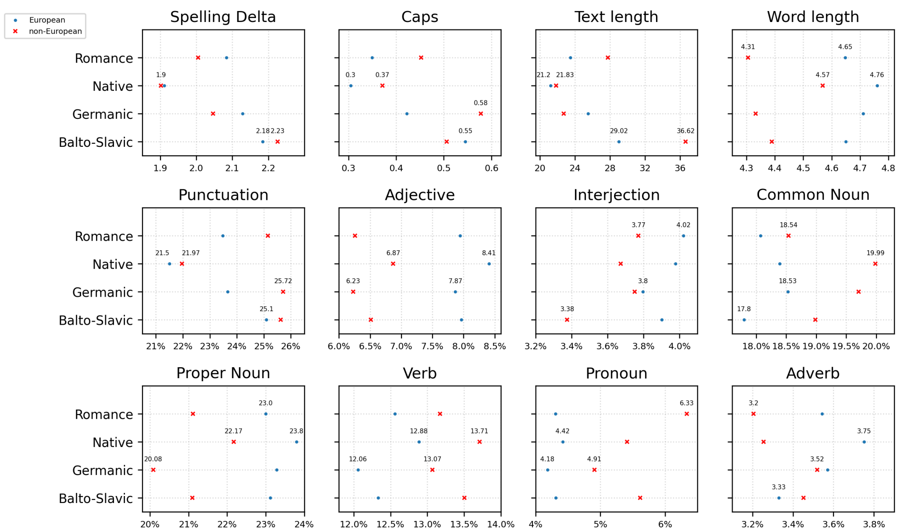

Figure 16: Average values of common language traits in Reddit dataset grouped by **Language Family**. Spelling delta, caps, text length and word length are average quantity values. Other values show average percentage of text which is that specific trait.

The trends mentioned before are reflected clearly in Figure 16. Scores for word length, adjectives, interjections and proper nouns are consistently higher in the European, with the lowest value in the European dataset being above the highest in the non-European. The same is true in reverse for common nouns, verbs and pronouns. The Balto-Slavic family also continues the trend of having the longest text length, with its lowest value (29.02) being higher than other language families. The spelling delta shows Native users with the lowest values, which is what we expected.

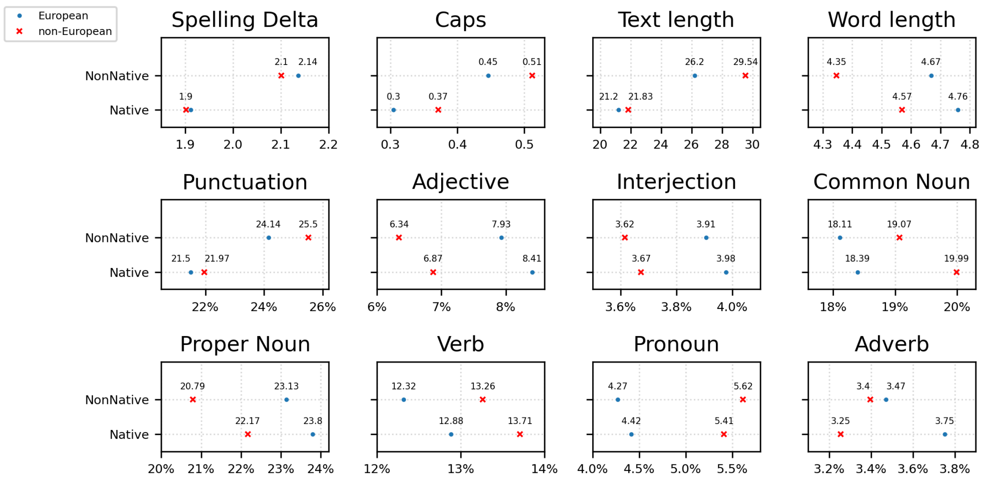

Figure 17: Average values of common language traits in Reddit dataset grouped by **Origin**. Spelling delta, caps, text length and word length are average quantity values. Other values show average percentage of text which is that specific trait.

While Native speakers have the lowest spelling delta, caps and text length in Figure 17, their words are on average 4.81% longer in European sub-reddits, and 1.89% in non-European. Trends which were already common in the *Language* and *Language Family* grouping continue to be present. Text in non-European sub-reddits is longer and contains more caps, punctuation, common nouns, verbs and pronouns, whereas text from European sub-reddits features longer words, a higher percentage of adjectives, interjections, proper nouns and adverbs.

	Character tri-grams	Word bi-grams	Word uni-grams	Parts-of-Speech bi-grams	Function words
Native	the, 3319 ent, 2688 and, 2580 ter, 2514 rea, 2416	i think, 306 i m, 279 it s, 204 i would, 157 i know, 119	i, 4131 would, 1425 peopl, 1423 like, 1276 countri, 1143	^^, 12336 ^N, 9747 N ^, 8956 N N, 8309 V ^, 7185	the, 12893 to, 7620 of, 6466 a, 6172 and, 5633
Germanic	the, 3187 ent, 2461 and, 2456 ter, 2385 rea, 2210	i think, 290 i m, 249 it s, 172 i would, 152 i know, 111	i, 3946 like, 1134 would, 988 peopl, 986 one, 902	^^, 11504 ^N, 8968 N N, 8449 N ^, 8174 G G, 6801	the, 10865 to, 6429 a, 5566 of, 5532 and, 5421
Balto-Slavic	ent, 3383 ter, 3318 the, 3143 rea, 3103 ian, 3079	rbc ru, 390 i think, 352 lenta ru, 229 ru news, 211 it s, 208	i, 5227 peopl, 1861 like, 1585 countri, 1454 one, 1181	^^, 15821 ^N, 11843 N ^, 11337 N N, 10260 G G, 8985	the, 12895 to, 8944 of, 7489 and, 7462 in, 6391
Romance	the, 2725 ent, 2392 rea, 2107 ter, 2076 ver, 1967	i think, 242 i m, 202 it s, 172 i would, 123 i know, 115	i, 3694 peopl, 1240 would, 1074 like, 1030 countri, 894	^^, 10809 ^N, 8173 N ^, 7537 N N, 7223 V ^, 6030	the, 11643 to, 6301 of, 5671 a, 5403 and, 4971
Total	the, 12374 ent, 10924 ter, 10293 rea, 9836 and, 9396	i think, 1190 i m, 948 it s, 756 i would, 588 i know, 491	i, 16998 people, 5510 like, 5025 would, 4665 countri, 4330	^^, 50470 ^N, 38731 N ^, 36004 N N, 34241 V ^, 27896	the, 48296 to, 29294 of, 25158 and, 23487 a, 23089

Table 20: Highest occurring n-gram values and their quantity in each language family from European Reddit data.

Character tri-grams from each language family are similar to the over-all Reddit data in Table 20, only Balto-Slavic has high variance in its ranking. This trend continues in word bi-grams, which includes specific terms such as *rbc ru*, *lenta ru* and *ru news*, which are all Russian media websites. We can also see a clear topic in both word bi-grams and uni-grams: expressing opinions. *I think*, *I would*, *I know* and more specifically the focus on **I** clearly states that users on Reddit are very keen on giving their personal input to various topics (in this case country related discussions e.g.*countri*). Other variances are the increased usage of foreign words (**G**) in both Germanic and Russian Parts-of-Speech bi-grams, most likely due to using native terms which have no English equivalents, and the low occurrence of the function word *a* in Balto-Slavic, which is related to the absence of definite and indefinite articles in e.g. Russian.

	Character tri-grams	Word bi-grams	Word uni-grams	Parts-of-Speech bi-grams	Function words
Native	the, 3233 ent, 2575 rea, 2555 thi, 2308 ver, 2292	i m, 836 it s, 507 i think, 456 i ve, 289 i d, 284	i, 6669 peopl, 1462 would, 1401 like, 1398 think, 1061	^^, 11753 ^N, 9860 N N, 9285 N ^, 8595 V N, 7495	the, 12416 to, 8279 a, 6861 of, 5583 and, 5212
Germanic	the, 2294 rea, 2186 ent, 1876 ter, 1765 com, 1707	i m, 491 it s, 480 i think, 292 i ve, 193 that s, 149	i, 4984 like, 1101 get, 967 one, 875 would, 847	N N, 8176 ^N, 7523 ^^, 7356 N ^, 6389 V N, 6010	the, 9769 to, 5723 a, 5645 and, 4402 is, 3931
Balto-Slavic	the, 4404 rea, 4328 ent, 3897 ter, 3786 oul, 3411	i m, 1417 it s, 628 i ve, 611 i think, 481 i ll, 328	i, 13396 like, 2449 would, 2357 one, 1864 get, 1706	^^, 16763 N N, 15110 ^N, 14712 N ^, 12819 V N, 12500	the, 17624 to, 12462 and, 10893 a, 10140 of, 8603
Romance	the, 2639 rea, 2624 ter, 2270 eve, 2024 com, 2015	i m, 1066 it s, 585 i think, 470 i ve, 299 that s, 231	i, 8280 like, 1721 one, 1243 would, 996 think, 893	^^, 10615 ^N, 9181 N N, 9027 N ^, 7947 V N, 7335	the, 11725 to, 6996 a, 6889 and, 6082 of, 4988
Total	the, 12570 rea, 11693 ent, 10271 ter, 10005 ver, 9269	i m, 3810 it s, 2200 i think, 1699 i ve, 1392 that s, 893	i, 33329 like, 6669 would, 5601 one, 4863 get, 4512	^^, 46487 N N, 41598 ^N, 41276 N ^, 35750 V N, 33340	the, 51534 to, 33460 a, 29535 and, 26589 of, 22811

Table 21: Highest occurring n-gram values and their quantity in each language family from non-European Reddit data.

Compared to the European data, we can find a different focus on personal expression in Table 21. Each language family has terms such as 'I am' and 'It is' at the top of word bi -and uni-grams instead of 'I think' and at higher quantity, with less specific context words (e.g. no 'country' but general opinion terms 'would', 'like'). Parts-of-Speech bi-grams for Germanic also show nouns as the most frequent with proper nouns only at third, which were first by a large margin in the European subreddit data. Surprisingly we can find the indefinite article 'a' in the functions words for Balto-Slavic.

5.2.3. Platform

Lastly, we compared the text features from Reddit and Twitter.

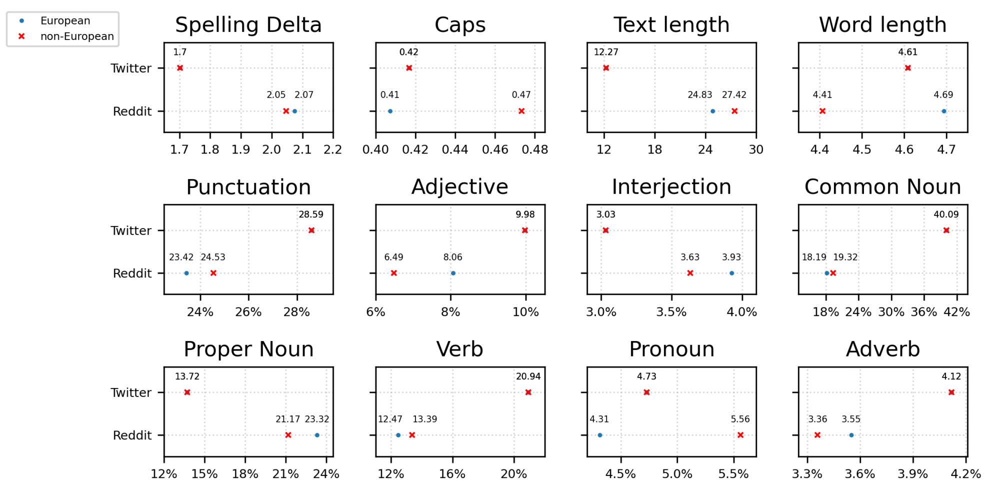

Figure 18: Average values of common language traits in both dataset grouped by **Platform**. Spelling delta, caps, text length and word length are average quantity values. Other values show average percentage of text which is that specific trait.

While Twitter shows a smaller spelling delta (1.7 compared to 2.05 and 2.07), their average text length is half of that in the Reddit dataset (12.27 average words versus 24.83 and 27.42). Text from Twitter contains more punctuation (28.59% of text), adjectives (9.98%), common nouns (40.09%, more than double that of 19.32%, the highest value in the Reddit dataset), verbs (20.94%) and adverbs (4.12%). Reddit data shows the highest difference in proper noun usage (21.17% and 23.32% to Twitters 13.72%), which is in line with its more topic focused discussions, most likely involving several high profile persons and places. Pronoun usage in non-European sub-reddits is also considerably higher (5.56% to 4.73%), strengthening the thesis that personal opinions and discussions are centric to Reddit.

6. Conclusion

We classified native language, language family and origin of texts from Twitter and Reddit. We developed a feature-set and implemented a model which can predict non-Native English speaking users from Twitter at more than 94% accuracy, their language family at 85%, native language at 66%, and the text category at 82%. We can also accurately distinguish Native and non-Native English speaking users from Reddit, however we did not meet baseline scores in all categories for the predictions. We observed outliers in language traits from non-European and European Reddit texts such as the averagely high text length from Russian users. Lastly, we compared the automated machine learning pipeline TPOT to traditional classification models and obtained equal or better results in most cases. In conclusion, we were able to create an automated machine learning pipeline and a feature-set which obtained very high percent results for our proposed task on the Twitter dataset, but failed to meet prediction scores from similar works such as Goldin et al.

As future work we will analyse word embedding techniques such as *Word2Vec*, which includes word context to connect similar text vectors. Most of the frequently used words in our datasets include context such as country-specific or language-specific text, which improves the probability of prediction. We also intend to make use of transformers such as BERT [Devlin et al., 2018] on language identification tasks. It pre-trains language models by using a "masked language model", which randomly masks some of the word tokens and creates an objective to correctly predict the original solely based on its context. Other than contextual features, Goldin et al. [Goldin et al., 2018] also proposed the use of platform-specific features. While they essentially bind the model to a specific source, it can narrow down trends for engagement metrics. These platform-specific or social-features can be used to employ transfer learning. Metrics on Twitter such as *Likes*, *Replies*, *At-mentions* etc. can be directly translated to Reddit in the form of *Upvotes*, *Comment-chain length* and *Reddit-mention*. Jun et al. proposed [Sun et al., 2016] a transfer learning procedure for predicting user roles in an unlabelled domain. Using a similar approach for origin, language family or language by transferring social-features from Twitter could improve some results.

List of Tables

List of Figures

Acronyms

API Application Programming Interface. 11

AutoML Automated Machine Learning. 2, 5, 6, 23, 26–28, 30–32

BERT Bidirectional Encoder Representations from Transformers. 50

COVID19 Coronavirus Disease 2019. 11

NASTY Nasty Advanced Search Tweet Yielder. 11, 13

NLP Natural Language Processing. 16

SMS Short Message Service. 11

TF-IDF Term frequency-inverse document frequency. 20, 22, 23, 25–29, 33–35, 51, 53

TOS Terms-of-Service. 11, 12

TPOT Tree-based Pipeline Optimization Tool. 6, 7, 23–26, 29, 32, 50

URL Uniform Resource Locator. 4, 13, 15, 19, 21, 22

Appendices

A. Tables

Feature	Language	Language Family	Origin	Category
elongated	5.28845	4.13815	2.16603	5.72438
caps	6.57517	6.22239	4.37856	7.63841
textLength	25.20496	20.21258	8.51775	19.37589
sentenceWordLength	37.60421	32.13924	15.40451	37.04075
spellDelta	39.85847	33.14391	14.46814	30.14507
#	42.91046	32.05827	9.61713	21.05902
@	0	0	0	0
E	3.51046	1.36481	1.36355	2.87964
,	29.47916	25.62539	12.86359	21.26438
~	1.48213	1.81707	1.42179	1.47579
U	19.5101	14.08161	4.02919	14.80755
A	18.88813	15.74507	7.54029	14.99583
D	9.18176	9.09453	5.22321	6.6851
!	8.20131	6.53738	3.5683	6.03088
N	30.15266	23.52981	12.10005	26.12002
P	11.75734	8.84296	5.041	8.89837
O	12.95857	11.01921	7.83614	10.2672
R	10.87107	9.38256	5.45222	10.50266
&	4.4454	4.57118	2.437	3.50954
L	0.97942	1.05193	1.12984	1.21822
Z	0.038	0	0.02931	0.04501
^	19.21098	16.13761	6.66986	17.81114
V	24.95992	20.43913	10.92249	18.58474
$	11.43838	10.18026	3.40628	9.06447
G	18.64962	15.61769	6.86786	11.94984
T	0.39486	0.76035	0.09581	0.41113
X	0.10357	0.10773	0.05999	0.12935
S	0.02004	0.03689	0	0.10246
Y	0	0	0	0
M	0	0	0	0

Table 22: Importance-feature data from classes in the Twitter dataset. Zero-values are marked in grey.

Feature	Language	Language Family	Origin	Category
charTrigrams_similarity_French	33.46309	28.0346	11.52262	31.43112
wordBigrams_similarity_French	25.40829	9.27151	3.03083	6.59921
wordUnigrams_similarity_French	53.08997	28.38437	13.83647	24.126
POSBigrams_similarity_French	20.73066	17.60291	9.76369	15.20972
functionWords_similarity_French	10.6487	8.71258	4.91319	8.23148
charTrigrams_similarity_German	37.83867	26.59925	11.30736	30.00291
wordBigrams_similarity_German	54.16398	20.5095	7.18534	8.82269
wordUnigrams_similarity_German	80.99097	43.47957	14.72136	23.01913
POSBigrams_similarity_German	21.17234	16.15745	9.29148	15.11584
functionWords_similarity_German	12.17971	10.08742	4.39057	6.88082
charTrigrams_similarity_Greek	35.8762	30.97146	13.0283	26.63953
wordBigrams_similarity_Greek	70.13044	27.66755	7.8467	5.978
wordUnigrams_similarity_Greek	66.95357	39.74756	17.66191	23.64423
POSBigrams_similarity_Greek	19.57383	16.89323	7.89692	14.66342
functionWords_similarity_Greek	12.14546	11.98288	7.73068	7.4566
charTrigrams_similarity_Indian	42.292	38.53892	15.1622	29.88334
wordBigrams_similarity_Indian	60.30732	59.81813	11.68101	14.3748
wordUnigrams_similarity_Indian	100	100	18.78624	30.58286
POSBigrams_similarity_Indian	20.34776	18.10354	7.66498	15.39668
functionWords_similarity_Indian	9.34233	7.3248	3.57302	8.04985
charTrigrams_similarity_Russian	34.77951	26.01636	10.53266	30.72254
wordBigrams_similarity_Russian	23.04427	8.14239	2.35191	7.6664
wordUnigrams_similarity_Russian	47.47752	25.52539	11.47915	30.85165
POSBigrams_similarity_Russian	24.1314	18.5984	8.76855	15.63061
functionWords_similarity_Russian	16.7963	12.46364	8.02677	12.06664
charTrigrams_similarity_Japanese	31.49123	26.5997	11.64725	27.10977
wordBigrams_similarity_Japanese	24.11335	25.27933	3.19493	7.5295
wordUnigrams_similarity_Japanese	42.75908	37.7734	9.96696	26.20745
POSBigrams_similarity_Japanese	24.09176	19.6695	10.6092	16.83545
functionWords_similarity_Japanese	15.34493	14.9748	7.83774	13.64005
charTrigrams_similarity_Turkish	33.1775	26.45136	10.66332	25.90176
wordBigrams_similarity_Turkish	57.78217	49.33136	6.66396	7.8799
wordUnigrams_similarity_Turkish	54.70823	52.71919	12.67926	21.69384
POSBigrams_similarity_Turkish	22.08994	15.51237	7.33839	14.36629
functionWords_similarity_Turkish	10.83582	8.76381	5.43933	8.26318
charTrigrams_similarity_Japonic	32.71036	29.02105	13.41398	28.50019
wordBigrams_similarity_Japonic	31.95974	32.30829	6.26926	8.05061
wordUnigrams_similarity_Japonic	41.19824	34.00177	11.49545	23.51003
POSBigrams_similarity_Japonic	25.03464	19.90445	8.90894	16.72097
functionWords_similarity_Japonic	16.25553	13.03247	6.43912	13.89024
charTrigrams_similarity_English	31.22214	27.9048	20.09066	29.55869
wordBigrams_similarity_English	74.82809	73.58554	100	6.40722
wordUnigrams_similarity_English	79.40175	81.52113	77.50227	26.75778
POSBigrams_similarity_English	20.82273	16.68885	9.18127	14.56049
functionWords_similarity_English	10.18973	8.49217	9.02823	8.33022
charTrigrams_similarity_Turkic	28.38197	24.03401	11.06656	27.84477
wordBigrams_similarity_Turkic	47.50505	50.56013	6.41171	7.24531
wordUnigrams_similarity_Turkic	54.92202	49.19992	16.6721	21.37334
POSBigrams_similarity_Turkic	19.46922	17.48535	7.94467	13.86221
functionWords_similarity_Turkic	9.65844	8.86985	4.26106	7.90544
charTrigrams_similarity_Indo-Aryan	43.82167	39.76729	15.24715	28.18929
wordBigrams_similarity_Indo-Aryan	72.92837	73.4299	12.02937	19.04242
wordUnigrams_similarity_Indo-Aryan	97.1119	96.26286	20.74448	26.5574
POSBigrams_similarity_Indo-Aryan	23.52593	19.25821	6.89365	13.18056
functionWords_similarity_Indo-Aryan	11.28792	9.08265	5.0223	7.22765
charTrigrams_similarity_Indo-European	33.70824	27.98327	14.95863	32.97203
wordBigrams_similarity_Indo-European	41.2221	55.29786	12.94005	11.62153
wordUnigrams_similarity_Indo-European	53.93787	64.60751	19.97353	23.79993
POSBigrams_similarity_Indo-European	19.13069	17.08819	6.98558	13.2862
functionWords_similarity_Indo-European	12.16154	10.35225	5.60462	7.37523
charTrigrams_similarity_Native	36.66508	33.86598	20.84167	25.39943
wordBigrams_similarity_Native	71.7624	64.82751	95.52166	6.80028
wordUnigrams_similarity_Native	84.15733	86.99338	85.55019	25.87281
POSBigrams_similarity_Native	22.96938	15.41677	7.85823	14.69431
functionWords_similarity_Native	11.80792	9.51967	5.142	8.06084
charTrigrams_similarity_NonNative	29.05696	24.65802	16.62919	29.71817
wordBigrams_similarity_NonNative	41.07483	44.59158	26.93741	25.25996
wordUnigrams_similarity_NonNative	49.42721	41.17208	29.30351	27.51903
POSBigrams_similarity_NonNative	22.12291	17.72151	7.6764	14.68859
functionWords_similarity_NonNative	10.40103	8.66704	5.09757	7.82242
charTrigrams_similarity_ArtCul	32.73137	27.1607	13.57878	32.71245
wordBigrams_similarity_ArtCul	34.82348	34.0603	5.29782	74.77449
wordUnigrams_similarity_ArtCul	38.10939	34.46757	13.51774	72.76252
POSBigrams_similarity_ArtCul	19.23489	16.16754	7.11793	14.29029
functionWords_similarity_ArtCul	10.60249	8.65795	4.13883	7.66788
charTrigrams_similarity_BuiTecSci	31.68823	26.61313	11.54911	45.07797
wordBigrams_similarity_BuiTecSci	8.39623	8.11539	4.74576	63.67829
wordUnigrams_similarity_BuiTecSci	30.37351	24.76748	11.32835	66.91548
POSBigrams_similarity_BuiTecSci	18.68178	17.40673	7.6898	14.86896
functionWords_similarity_BuiTecSci	10.08861	8.42247	5.60122	8.12085
charTrigrams_similarity_Pol	29.57745	24.07578	13.17869	47.29456
wordBigrams_similarity_Pol	9.88788	7.91351	6.74782	79.2082
wordUnigrams_similarity_Pol	33.08153	27.755	17.39274	100
POSBigrams_similarity_Pol	20.11796	15.77703	7.66243	14.03557
functionWords_similarity_Pol	10.84694	8.95381	7.03807	8.2155
charTrigrams_similarity_SocSoc	29.88514	25.60531	13.16307	37.88883
wordBigrams_similarity_SocSoc	9.5633	8.63615	5.37838	55.21921
wordUnigrams_similarity_SocSoc	28.64861	25.18573	13.62436	54.54032
POSBigrams_similarity_SocSoc	20.86587	16.18752	7.10903	14.16224
functionWords_similarity_SocSoc	10.33784	8.54556	5.34974	6.85409

Table 23: Importance-feature data for n-gram similarity from classes in the Twitter dataset.

Feature	Language	Language Family	Origin
elongated	9.24245	8.67596	7.40112
caps	16.29179	13.92986	12.79495
textLength	50.39557	46.20926	39.94194
sentenceWordLength	92.25562	89.6638	83.85397
spellDelta	100	100	100
#	1.06408	0.71106	0.84253
@	0	0	0
E	0	0	0
,	71.17361	68.03948	65.38476
~	0	0	0
U	0	0	0
A	56.33383	55.13073	50.80933
D	17.72815	17.07164	19.72544
!	43.18374	41.38549	38.77613
N	69.30895	67.38842	64.72361
P	16.56145	15.67545	20.02514
O	46.06746	42.88158	37.38148
R	41.97623	38.88358	39.54107
&	15.62777	13.48878	13.40582
L	0	0	0
Z	0	0	0
^	72.64881	71.66134	65.15304
V	66.40304	64.41854	56.05199
$	19.24085	17.87101	17.60662
G	38.79284	35.04991	31.74181
T	12.18522	10.43491	10.24263
X	0	0	0
S	0	0	0
Y	0	0	0
M	0	0	0

Table 24: Importance-feature data from classes in the Reddit European dataset. Zero-values are marked in grey.

Feature	Language	Language Family	Origin
charTrigrams_similarity_French	58.10987	57.55943	54.62284
wordBigrams_similarity_French	27.82606	26.01588	18.9955
wordUnigrams_similarity_French	57.49862	56.54965	48.83289
POSBigrams_similarity_French	26.73172	24.57464	20.97586
functionWords_similarity_French	24.33316	19.63374	16.52943
charTrigrams_similarity_German	58.85734	58.32516	54.36286
wordBigrams_similarity_German	26.75079	23.73461	18.86263
wordUnigrams_similarity_German	57.05698	56.04041	45.94905
POSBigrams_similarity_German	27.89204	24.6747	20.53243
functionWords_similarity_German	24.07204	18.74638	16.91302
charTrigrams_similarity_Russian	69.05211	67.28428	62.36563
wordBigrams_similarity_Russian	37.76267	28.535	18.18266
wordUnigrams_similarity_Russian	78.06539	73.61069	61.82648
POSBigrams_similarity_Russian	27.21622	23.90559	20.15029
functionWords_similarity_Russian	24.20607	20.49079	16.76616
charTrigrams_similarity_Bulgarian	62.37298	62.9625	56.5986
wordBigrams_similarity_Bulgarian	30.99107	27.49927	18.9921
wordUnigrams_similarity_Bulgarian	65.60022	62.30061	50.27557
POSBigrams_similarity_Bulgarian	27.22175	23.90045	21.47844
functionWords_similarity_Bulgarian	24.93449	21.17097	18.21671
charTrigrams_similarity_Croatian	61.15692	59.34397	53.96762
wordBigrams_similarity_Croatian	25.01819	21.66508	18.06934
wordUnigrams_similarity_Croatian	56.45351	56.23054	47.41347
POSBigrams_similarity_Croatian	26.93851	25.09519	20.49457
functionWords_similarity_Croatian	24.03472	20.39093	16.45351
charTrigrams_similarity_Czech	58.33817	58.24675	53.53689
wordBigrams_similarity_Czech	26.94518	24.95422	19.89355
wordUnigrams_similarity_Czech	54.30984	53.41105	44.60831
POSBigrams_similarity_Czech	27.38302	25.86126	19.91583
functionWords_similarity_Czech	24.29238	19.19956	17.01004
charTrigrams_similarity_Lithuanian	60.70204	59.55282	54.61577
wordBigrams_similarity_Lithuanian	25.5146	21.6152	17.17686
wordUnigrams_similarity_Lithuanian	57.34353	57.16921	46.33432
POSBigrams_similarity_Lithuanian	27.95806	24.32686	20.58725
functionWords_similarity_Lithuanian	23.37908	19.59277	15.93205
charTrigrams_similarity_Polish	59.13419	59.05656	57.05848
wordBigrams_similarity_Polish	25.42742	21.55783	16.90266
wordUnigrams_similarity_Polish	59.91373	58.31284	50.41692
POSBigrams_similarity_Polish	28.71877	26.12245	20.82237
functionWords_similarity_Polish	23.86443	19.72574	16.79427
charTrigrams_similarity_Serbian	66.20222	66.08626	63.91905
wordBigrams_similarity_Serbian	31.80988	24.30585	16.97577
wordUnigrams_similarity_Serbian	69.71719	67.15322	55.27423
POSBigrams_similarity_Serbian	27.2295	25.1137	21.98849
functionWords_similarity_Serbian	24.47296	19.88535	17.34201
charTrigrams_similarity_Slovene	57.76847	58.11649	53.77295
wordBigrams_similarity_Slovene	24.73977	21.93869	18.38523
wordUnigrams_similarity_Slovene	56.62839	53.80898	47.36068
POSBigrams_similarity_Slovene	27.49014	24.68222	20.34127
functionWords_similarity_Slovene	24.17919	19.61308	16.92485
charTrigrams_similarity_Finnish	56.80678	54.88089	53.68858
wordBigrams_similarity_Finnish	25.2352	23.05903	18.11583
wordUnigrams_similarity_Finnish	54.48461	53.91107	46.16477
POSBigrams_similarity_Finnish	27.96435	23.64975	19.81638
functionWords_similarity_Finnish	23.63229	18.62002	16.35914
charTrigrams_similarity_Dutch	59.14561	60.31068	55.0559
wordBigrams_similarity_Dutch	25.58626	21.97396	18.81561
wordUnigrams_similarity_Dutch	58.53122	58.42889	47.84956
POSBigrams_similarity_Dutch	27.34424	24.33997	20.70933
functionWords_similarity_Dutch	23.00879	19.71687	16.16999
charTrigrams_similarity_Norwegian	58.17293	57.04304	54.37937
wordBigrams_similarity_Norwegian	25.91306	22.76188	18.79988
wordUnigrams_similarity_Norwegian	57.10179	57.54462	45.81272
POSBigrams_similarity_Norwegian	27.89929	24.48582	19.98754
functionWords_similarity_Norwegian	24.08657	19.5853	17.24719
charTrigrams_similarity_Swedish	62.90033	64.20957	57.81889
wordBigrams_similarity_Swedish	29.09722	26.58382	18.67052
wordUnigrams_similarity_Swedish	64.28909	66.87721	50.82412
POSBigrams_similarity_Swedish	27.47519	24.78716	20.63061
functionWords_similarity_Swedish	24.13796	20.17464	16.61181
charTrigrams_similarity_Italian	56.79584	55.88619	51.80526
wordBigrams_similarity_Italian	26.70241	22.89343	18.75443
wordUnigrams_similarity_Italian	54.4695	54.70749	43.85598
POSBigrams_similarity_Italian	27.54401	24.83183	19.83848
functionWords_similarity_Italian	24.50645	20.00097	16.41333
charTrigrams_similarity_Spanish	57.68063	57.93391	51.83597
wordBigrams_similarity_Spanish	26.877	24.16216	18.60974
wordUnigrams_similarity_Spanish	57.93193	55.3952	46.37725
POSBigrams_similarity_Spanish	26.55689	24.65745	21.13495
functionWords_similarity_Spanish	23.80578	19.65393	17.00933
charTrigrams_similarity_Portugese	58.85472	56.25309	52.04699
wordBigrams_similarity_Portugese	24.6302	22.83006	17.35433
wordUnigrams_similarity_Portugese	53.24149	53.00798	44.99736
POSBigrams_similarity_Portugese	26.73911	25.10392	20.00822
functionWords_similarity_Portugese	24.30572	19.08077	16.20287
charTrigrams_similarity_Romanian	57.61731	58.3944	53.38843
wordBigrams_similarity_Romanian	24.73757	22.73672	16.19954
wordUnigrams_similarity_Romanian	56.31631	55.32376	45.76543
POSBigrams_similarity_Romanian	27.46879	24.27392	20.52131
functionWords_similarity_Romanian	24.14544	19.4614	17.14482
charTrigrams_similarity_Balto-Slavic	55.80357	56.42204	50.17539
wordBigrams_similarity_Balto-Slavic	28.31224	28.6514	19.81223
wordUnigrams_similarity_Balto-Slavic	52.59421	57.0678	44.43495
POSBigrams_similarity_Balto-Slavic	28.05185	24.53689	20.09978
functionWords_similarity_Balto-Slavic	23.03003	19.45843	16.30989
charTrigrams_similarity_Germanic	55.59816	55.35856	51.53454
wordBigrams_similarity_Germanic	24.04033	24.26415	20.3591
wordUnigrams_similarity_Germanic	50.83475	56.16022	41.7777
POSBigrams_similarity_Germanic	26.89372	24.00695	21.35709
functionWords_similarity_Germanic	23.85114	19.1926	16.73414
charTrigrams_similarity_Romance	52.84264	51.75028	48.92293
wordBigrams_similarity_Romance	24.03357	23.93329	19.16818
wordUnigrams_similarity_Romance	50.53202	51.02924	40.82107
POSBigrams_similarity_Romance	27.71171	24.84241	20.69924
functionWords_similarity_Romance	22.84106	19.51971	16.99006
charTrigrams_similarity_English	53.32087	52.15612	52.70691
wordBigrams_similarity_English	26.59448	26.52103	32.88305
wordUnigrams_similarity_English	54.18819	55.24387	55.51887
POSBigrams_similarity_English	27.95463	23.25505	20.6857
functionWords_similarity_English	24.06428	19.10393	16.47077
charTrigrams_similarity_Native	53.42903	51.63233	50.2551
wordBigrams_similarity_Native	26.6356	28.33614	32.67336
wordUnigrams_similarity_Native	53.53162	53.474	56.20856
POSBigrams_similarity_Native	27.21454	23.68467	20.17628
functionWords_similarity_Native	23.83018	19.32449	16.66134
charTrigrams_similarity_NonNative	51.14872	49.72704	49.44056
wordBigrams_similarity_NonNative	24.17457	22.48155	20.29898
wordUnigrams_similarity_NonNative	49.16173	47.09122	42.94739
POSBigrams_similarity_NonNative	27.72949	24.07848	19.76905
functionWords_similarity_NonNative	23.69275	19.17212	17.45123
charTrigrams_similarity_Reddit	51.19171	50.05473	47.53265
wordBigrams_similarity_Reddit	24.63404	21.29172	18.5487
wordUnigrams_similarity_Reddit	45.66101	45.96253	39.50629
POSBigrams_similarity_Reddit	27.04044	24.66167	19.84932
functionWords_similarity_Reddit	23.86676	19.04944	16.7046

Table 25: Importance-feature data for n-gram similarity from classes in the Reddit European dataset.

Feature	Language	Language Family	Origin
elongated	9.65072	10.17834	6.74281
caps	19.44495	17.08424	14.61497
textLength	50.25479	50.08625	41.52016
sentenceWordLength	100	100	100
spellDelta	87.56136	92.80434	92.30004
#	0.9641	0.93342	0.69994
@	0	0	0
E	0	0	0
,	77.83003	73.85998	73.88103
~	0	0	0
U	0	0	0
A	47.53202	47.57491	42.36091
D	13.29746	11.65754	13.25725
!	38.28913	38.2168	34.96804
N	65.37013	67.31084	57.32832
P	17.48906	17.05691	18.47218
O	45.50531	45.03284	37.16751
R	36.59429	37.07899	32.33865
&	13.76523	12.20176	10.63435
L	0	0	0
Z	0	0	0
^	70.92183	71.05572	58.67208
V	61.06454	59.91127	52.84259
$	20.82762	19.06745	16.80601
G	41.15973	40.56949	34.25042
T	19.41671	20.61859	20.6031
X	0	0	0
S	0	0	0
Y	0	0	0
M	0	0	0

Table 26: Importance-feature data from classes in the Reddit non-European dataset. Zero-values are marked in grey.

Feature	Language	Language Family	Origin
charTrigrams_similarity_French	56.16091	58.78767	50.89271
wordBigrams_similarity_French	30.07555	30.71104	21.89513
wordUnigrams_similarity_French	57.71274	59.61826	46.74384
POSBigrams_similarity_French	26.07667	24.261	20.59911
functionWords_similarity_French	22.07496	18.73277	14.79964
charTrigrams_similarity_German	59.5153	59.65776	49.98822
wordBigrams_similarity_German	31.11371	31.3242	24.96311
wordUnigrams_similarity_German	70.56261	69.50347	60.39894
POSBigrams_similarity_German	27.69091	25.9086	20.1971
functionWords_similarity_German	21.64593	19.43805	15.00689
charTrigrams_similarity_Russian	57.3505	59.21167	49.73687
wordBigrams_similarity_Russian	29.531	28.65785	20.92313
wordUnigrams_similarity_Russian	55.73446	57.80421	46.99016
POSBigrams_similarity_Russian	27.3709	25.66064	21.16696
functionWords_similarity_Russian	21.23926	19.15149	14.80942
charTrigrams_similarity_Bulgarian	62.2877	68.58882	56.47453
wordBigrams_similarity_Bulgarian	37.54677	32.62391	25.15637
wordUnigrams_similarity_Bulgarian	72.05086	71.31615	63.7065
POSBigrams_similarity_Bulgarian	27.01293	24.30081	18.63478
functionWords_similarity_Bulgarian	21.81692	19.0201	15.23336
charTrigrams_similarity_Croatian	55.88409	59.51005	47.66107
wordBigrams_similarity_Croatian	33.61748	30.21291	21.58082
wordUnigrams_similarity_Croatian	61.90671	58.23543	46.65595
POSBigrams_similarity_Croatian	26.81924	25.6368	20.60303
functionWords_similarity_Croatian	22.28749	18.69682	14.97385
charTrigrams_similarity_Czech	54.01511	56.27282	48.43931
wordBigrams_similarity_Czech	26.53546	24.72965	21.01364
wordUnigrams_similarity_Czech	51.46962	52.00583	42.73831
POSBigrams_similarity_Czech	27.40875	24.29652	20.14062
functionWords_similarity_Czech	21.68565	18.65757	15.52531
charTrigrams_similarity_Lithuanian	61.494	63.95467	55.14402
wordBigrams_similarity_Lithuanian	32.35784	31.17002	23.64432
wordUnigrams_similarity_Lithuanian	65.36356	63.53935	59.83854
POSBigrams_similarity_Lithuanian	27.02406	25.05663	20.20998
functionWords_similarity_Lithuanian	23.65606	19.17099	15.37815
charTrigrams_similarity_Polish	56.8012	58.92372	48.17989
wordBigrams_similarity_Polish	33.56108	32.59465	21.76825
wordUnigrams_similarity_Polish	66.64464	64.99992	47.99817
POSBigrams_similarity_Polish	26.51486	25.17034	21.34935
functionWords_similarity_Polish	21.63231	18.34056	15.34302
charTrigrams_similarity_Serbian	55.43461	58.64817	47.98209
wordBigrams_similarity_Serbian	33.97248	30.72229	22.77428
wordUnigrams_similarity_Serbian	61.56191	59.25039	46.00999
POSBigrams_similarity_Serbian	26.13691	24.49808	20.04466
functionWords_similarity_Serbian	21.65479	18.41401	14.30812
charTrigrams_similarity_Slovene	53.26177	57.64551	46.11048
wordBigrams_similarity_Slovene	27.95868	26.87708	21.9622
wordUnigrams_similarity_Slovene	51.73285	51.5432	45.51927
POSBigrams_similarity_Slovene	26.93983	25.03757	18.99527
functionWords_similarity_Slovene	21.74534	18.78121	14.57341
charTrigrams_similarity_Finnish	57.7288	57.35579	47.47474
wordBigrams_similarity_Finnish	32.93541	31.11326	23.94906
wordUnigrams_similarity_Finnish	65.091	63.20627	47.68559
POSBigrams_similarity_Finnish	26.88216	25.35279	19.50008
functionWords_similarity_Finnish	21.35708	19.36249	14.91397
charTrigrams_similarity_Dutch	56.5213	57.17528	48.61735
wordBigrams_similarity_Dutch	31.1924	28.82348	21.46292
wordUnigrams_similarity_Dutch	57.41996	57.79197	48.88929
POSBigrams_similarity_Dutch	26.46655	23.90803	18.73567
functionWords_similarity_Dutch	21.34401	19.66823	14.38865
charTrigrams_similarity_Norwegian	55.22002	56.13321	48.57194
wordBigrams_similarity_Norwegian	30.96362	29.74474	21.53629
wordUnigrams_similarity_Norwegian	63.11514	60.15699	43.91958
POSBigrams_similarity_Norwegian	26.19538	23.61816	20.52856
functionWords_similarity_Norwegian	21.46979	18.88871	14.80503
charTrigrams_similarity_Swedish	53.85736	54.17467	46.95639
wordBigrams_similarity_Swedish	28.10317	26.36002	20.49877
wordUnigrams_similarity_Swedish	53.42087	54.46732	48.12948
POSBigrams_similarity_Swedish	26.20449	24.19881	19.51232
functionWords_similarity_Swedish	22.14702	19.52832	16.0081
charTrigrams_similarity_Italian	54.11453	55.65267	47.91256
wordBigrams_similarity_Italian	29.55312	28.13453	21.30968
wordUnigrams_similarity_Italian	56.1544	52.77956	41.89788
POSBigrams_similarity_Italian	26.48577	24.66584	19.78411
functionWords_similarity_Italian	21.61009	19.01726	14.07609
charTrigrams_similarity_Spanish	57.74555	58.56779	48.83092
wordBigrams_similarity_Spanish	34.75372	32.69556	21.23902
wordUnigrams_similarity_Spanish	67.97554	63.07377	48.64655
POSBigrams_similarity_Spanish	26.26948	24.59751	19.95776
functionWords_similarity_Spanish	21.60967	18.87499	14.93553
charTrigrams_similarity_Portugese	54.35364	56.86596	48.84829
wordBigrams_similarity_Portugese	29.56563	28.93933	22.91967
wordUnigrams_similarity_Portugese	56.37941	56.82861	47.34344
POSBigrams_similarity_Portugese	25.82125	24.80233	20.06345
functionWords_similarity_Portugese	21.9949	19.46664	14.94154
charTrigrams_similarity_Romanian	53.65221	53.77123	46.89767
wordBigrams_similarity_Romanian	30.66622	31.93349	22.86451
wordUnigrams_similarity_Romanian	60.38428	60.62216	43.44268
POSBigrams_similarity_Romanian	26.34609	24.57416	19.63926
functionWords_similarity_Romanian	21.85322	17.34704	14.93256
charTrigrams_similarity_Balto-Slavic	49.56776	53.03728	43.65199
wordBigrams_similarity_Balto-Slavic	27.36672	28.14801	21.80075
wordUnigrams_similarity_Balto-Slavic	48.78912	54.00139	40.79321
POSBigrams_similarity_Balto-Slavic	26.21286	24.62504	18.95724
functionWords_similarity_Balto-Slavic	21.44993	19.28325	15.1387
charTrigrams_similarity_Germanic	50.11601	52.50757	46.99581
wordBigrams_similarity_Germanic	26.86455	26.90164	22.9035
wordUnigrams_similarity_Germanic	50.96339	56.39922	48.22701
POSBigrams_similarity_Germanic	26.43583	24.5344	20.1952
functionWords_similarity_Germanic	20.46407	18.79665	13.88024
charTrigrams_similarity_Romance	49.89896	51.49951	44.5921
wordBigrams_similarity_Romance	29.1987	30.92398	21.66013
wordUnigrams_similarity_Romance	52.4507	56.01687	45.61511
POSBigrams_similarity_Romance	25.7598	25.16614	20.35056
functionWords_similarity_Romance	21.2881	18.36043	15.09872
charTrigrams_similarity_English	53.48339	55.05224	51.60557
wordBigrams_similarity_English	33.00429	35.33038	40.59056
wordUnigrams_similarity_English	60.95645	67.60359	84.42672
POSBigrams_similarity_English	25.73419	24.44758	19.63076
functionWords_similarity_English	21.46689	18.0269	15.02177
charTrigrams_similarity_Native	54.03989	56.31122	52.86479
wordBigrams_similarity_Native	32.40228	36.33611	42.12138
wordUnigrams_similarity_Native	62.50696	68.81051	85.29703
POSBigrams_similarity_Native	26.5336	24.4171	18.52185
functionWords_similarity_Native	20.82645	18.74451	14.7901
charTrigrams_similarity_NonNative	47.57292	50.15008	42.15036
wordBigrams_similarity_NonNative	27.13365	25.29416	21.25017
wordUnigrams_similarity_NonNative	46.75531	49.07749	40.91646
POSBigrams_similarity_NonNative	26.18604	25.27421	19.40367
functionWords_similarity_NonNative	21.67307	18.41165	14.39382
charTrigrams_similarity_Reddit	48.42925	51.2632	41.71695
wordBigrams_similarity_Reddit	26.58375	25.83126	21.8308
wordUnigrams_similarity_Reddit	45.00007	44.8243	39.81778
POSBigrams_similarity_Reddit	25.78321	24.63724	18.2393
functionWords_similarity_Reddit	21.31984	18.8982	14.92622

Table 27: Importance-feature data for n-gram similarity from classes in the Reddit non-European dataset.

Language	elongated	caps	textLength	sentenceWordLength	spellDelta	#	E	,	~	U	A	D	!	N	P	O	R	&	L	^	V	$	G	T
English	0.137112	0.556673	14.58044	4.604157	1.743666	2.281536	0.005462	0.330499	0.003134	0.869287	0.121865	0.035769	0.01733	0.457594	0.037368	0.067971	0.052902	0.014338	0.003134	0.132356	0.251502	0.034518	0.08066	0.000632
French	0.146667	0.26	9.603333	4.705687	1.578164	2.706667	0.006723	0.341732	0.002531	0.823333	0.136405	0.041025	0.034327	0.501309	0.044814	0.053961	0.052522	0.009179	0.002613	0.158762	0.216637	0.035971	0.088931	0.000476
German	0.13289	0.32392	10.6196	4.532355	1.683908	2.501661	0.018055	0.385919	0.006593	0.848837	0.13081	0.03601	0.02355	0.479202	0.03277	0.053718	0.052279	0.010174	0.001855	0.151106	0.260499	0.072904	0.111456	0.000174
Greek	0.152961	0.355263	11.14638	4.484044	1.790254	2.386513	0.003829	0.360903	0.0015	0.712171	0.12511	0.029826	0.015739	0.481177	0.039192	0.043368	0.053285	0.01229	0.002296	0.14831	0.231636	0.04981	0.108546	0.001398
Indian	0.097426	0.368566	17.23254	4.831813	1.877417	2.014706	0.001943	0.252617	0.001768	0.828125	0.10228	0.027407	0.007757	0.394816	0.041706	0.033855	0.037671	0.01527	0.001626	0.183027	0.217393	0.034812	0.122384	0.00042
Japanese	0.127072	0.403315	5.18232	4.466367	1.232805	2.132597	0.015078	0.646191	0.012891	0.309392	0.154587	0.055938	0.073568	0.532037	0.068956	0.095996	0.059837	0.014733	0.003683	0.234012	0.437061	0.077778	0.10466	0.011971
Russian	0.168605	0.674419	9.122093	3.940442	1.642226	2.127907	0.003579	0.340816	0.002907	0.686047	0.092321	0.017819	0.043218	0.375384	0.021671	0.106551	0.076892	0.028088	0.003198	0.17957	0.316435	0.039407	0.130449	0
Turkish	0.142584	0.436364	9.026794	4.598399	1.589726	3.028708	0.010391	0.423955	0.006661	0.843062	0.161695	0.040686	0.053701	0.632242	0.046331	0.068226	0.065339	0.015387	0.003142	0.21532	0.325803	0.080888	0.143642	0.002334
Family																								
Indo-European	0.097426	0.368566	17.23254	4.831813	1.877417	2.014706	0.001943	0.252617	0.001768	0.828125	0.10228	0.027407	0.007757	0.394816	0.041706	0.033855	0.037671	0.01527	0.001626	0.183027	0.217393	0.034812	0.122384	0.00042
Indo-Aryan	0.146254	0.359691	10.47562	4.485278	1.699227	2.458383	0.009411	0.364383	0.003651	0.77824	0.125812	0.032809	0.02466	0.473243	0.036105	0.055423	0.055203	0.012593	0.002287	0.154372	0.247963	0.054543	0.108329	0.000653
Japonic	0.127072	0.403315	5.18232	4.466367	1.232805	2.132597	0.015078	0.646191	0.012891	0.309392	0.154587	0.055938	0.073568	0.532037	0.068956	0.095996	0.059837	0.014733	0.003683	0.234012	0.437061	0.077778	0.10466	0.011971
Native	0.137112	0.556673	14.58044	4.604157	1.743666	2.281536	0.005462	0.330499	0.003134	0.869287	0.121865	0.035769	0.01733	0.457594	0.037368	0.067971	0.052902	0.014338	0.003134	0.132356	0.251502	0.034518	0.08066	0.000632
Turkic	0.142584	0.436364	9.026794	4.598399	1.589726	3.028708	0.010391	0.423955	0.006661	0.843062	0.161695	0.040686	0.053701	0.632242	0.046331	0.068226	0.065339	0.015387	0.003142	0.21532	0.325803	0.080888	0.143642	0.002334
Category																								
ArtCul	0.137586	0.396415	11.72694	4.457934	1.642402	2.137059	0.006604	0.365304	0.003602	0.740643	0.126392	0.036326	0.030703	0.443596	0.038319	0.065218	0.054761	0.015879	0.002406	0.186661	0.270855	0.053738	0.108066	0.001537
BuiTecSci	0.090814	0.424843	13.23069	5.055125	1.896854	2.837161	0.00646	0.29363	0.005396	0.919624	0.119613	0.024213	0.013727	0.458973	0.038235	0.037141	0.038274	0.011085	0.001382	0.150859	0.233256	0.049607	0.109655	0.000314
Pol	0.139432	0.555062	14.37567	4.718402	1.757688	2.556838	0.005157	0.332791	0.002206	0.768206	0.120217	0.035395	0.025443	0.529059	0.044987	0.053519	0.057569	0.014277	0.002443	0.179447	0.285583	0.047965	0.110124	0.000924
SocSoc	0.15239	0.324617	10.44454	4.363848	1.6057	2.455365	0.011709	0.415056	0.005681	0.853922	0.147385	0.04041	0.034506	0.546651	0.043197	0.066298	0.059018	0.013893	0.003926	0.153537	0.261023	0.056369	0.124982	0.00234
Origin																								
Native	0.137112	0.556673	14.58044	4.604157	1.743666	2.281536	0.005462	0.330499	0.003134	0.869287	0.121865	0.035769	0.01733	0.457594	0.037368	0.067971	0.052902	0.014338	0.003134	0.132356	0.251502	0.034518	0.08066	0.000632
NonNative	0.131131	0.384134	11.6967	4.608356	1.69798	2.471972	0.007891	0.362296	0.004344	0.787538	0.130092	0.034446	0.029867	0.496133	0.041792	0.054736	0.05329	0.014149	0.002394	0.18172	0.268561	0.057113	0.121224	0.001542

Table 28: Average feature value for each class in Twitter dataset

Language	elongated	caps	textLength	sentenceWordLength	spellDelta	#	,	A	D	!	N	P	O	R	&	^	V	$	G	T
Bulgarian	0.060048	0.289832	31.16894	4.883648	2.018323	0.002402	17.34322	8.542345	0.731401	3.559926	19.39377	0.873064	5.449149	3.693336	0.785064	24.13455	15.20688	0.99437	3.474868	0.966783
Croatian	0.145032	0.39984	24.35737	4.660439	2.163586	0.003205	24.70376	8.203219	0.55003	4.064449	17.04849	0.590457	4.473404	3.57511	0.73737	23.9142	11.2231	0.97749	5.96441	0.460482
Czech	0.120192	0.446314	30.95513	4.64643	2.160226	0.008013	23.29782	8.578737	0.590263	3.986781	17.57847	0.490189	4.476604	3.592615	0.553558	22.94995	12.74086	1.318517	5.750139	0.372409
Dutch	0.112257	0.296117	27.89684	4.619945	2.041332	0.005461	22.72803	8.017379	0.626323	4.354696	17.25221	0.57116	4.889361	3.807629	0.687865	24.0875	11.94199	0.915863	5.710072	0.464561
English	0.091437	0.304522	21.19858	4.759762	1.911378	0.003401	21.50109	8.409533	0.782433	3.977332	18.39304	0.648554	4.417456	3.751195	0.565291	23.79617	12.88496	0.995259	4.972772	0.448902
Finnish	0.105518	0.342025	22.57307	4.745754	2.032912	0.007884	23.14949	7.82559	0.717041	3.891386	18.84381	0.709785	4.344491	3.671025	0.504963	22.54167	12.7663	0.976486	5.939579	0.398747
French	0.182535	0.28866	26.16252	4.672089	2.04382	0.004851	22.44813	7.937545	0.543729	3.929572	19.0226	0.574394	3.943077	3.66906	0.593459	22.66073	12.82317	1.010954	7.085529	0.446751
German	0.170406	0.42268	24.18375	4.92358	2.295091	0.007884	22.68985	7.811907	0.534112	3.185055	19.56162	0.644988	3.544361	3.908643	0.812074	23.87712	12.09635	1.222139	8.081874	0.511273
Italian	0.151607	0.377805	25.31837	4.617046	2.12471	0.011522	23.21127	7.682238	0.682638	4.45356	17.48221	0.499975	4.605074	3.384391	0.801907	22.71228	13.0635	1.082203	6.640879	0.557788
Lithuanian	0.063251	0.267414	20.85829	4.460364	2.071809	0.006405	26.0603	8.377293	0.437726	4.313001	17.6598	0.735222	4.381958	3.763132	0.537887	20.82796	12.07055	1.216004	5.471679	0.516517
Norwegian	0.070346	0.303214	19.20558	4.622724	1.921545	0.006671	22.76286	8.239785	0.805978	4.130616	18.72619	0.569865	4.606412	3.168655	0.612351	22.68778	12.78971	1.383503	5.504638	0.458852
Polish	0.092874	0.368295	21.67654	4.535238	2.189167	0.015212	28.91052	7.67466	0.843224	3.560122	16.39836	0.558591	4.523544	2.813044	0.485616	22.34726	11.92594	1.090928	7.377191	0.389275
Portuguese	0.132201	0.358399	23.0188	4.715597	2.086646	0.005458	23.04167	7.419932	0.745667	3.623761	18.28804	0.693281	4.087978	3.62437	0.553996	23.95942	12.614	1.214151	6.442921	0.465037
Romanian	0.130988	0.385688	22.75622	4.550456	2.154022	0.003032	26.38627	8.374136	0.747312	4.24009	17.09251	0.583089	4.403889	3.663824	0.742979	21.94305	11.10608	0.994402	6.003957	0.575967
Russian	0.117694	1.684548	59.84067	4.826508	2.747844	0.079263	30.56071	7.260064	0.195751	2.651715	19.4027	0.787915	2.851434	2.635162	0.948192	23.93847	10.56693	1.084519	11.5162	0.613719
Serbian	0.082466	0.622898	21.35869	4.707194	2.065964	0.01281	24.88821	7.641022	0.568027	4.759334	17.32278	0.521064	3.644054	3.338372	0.641083	24.52556	12.81725	1.191183	4.186554	0.207803
Slovene	0.116894	0.284227	21.96797	4.47432	2.060159	0.004003	25.01716	7.426163	0.613341	4.317668	17.58044	0.731176	4.711441	3.229589	0.74136	22.33246	12.11222	1.137828	6.573289	0.401729
Spanish	0.087379	0.336165	20.14806	4.683857	2.008046	0.010922	22.29112	8.302608	0.61249	3.858419	18.47975	0.609373	4.488653	3.375498	0.593479	23.71602	13.19292	0.942221	4.708413	0.31816
Swedish	0.120679	0.747726	33.60582	4.640625	2.352836	0.012129	26.9548	7.44698	0.891225	3.41813	18.25628	0.70495	3.528229	3.298897	0.643659	23.23284	10.70235	1.625168	8.022312	0.388702
Family																				
Balto-Slavic	0.0998	0.545445	29.02322	4.649267	2.184639	0.016416	25.09793	7.962852	0.56622	3.9016	17.7982	0.660984	4.313916	3.329994	0.678773	23.12124	12.33304	1.12635	6.289378	0.491105
Germanic	0.115842	0.422368	25.49272	4.710537	2.128754	0.008006	23.65712	7.86831	0.714946	3.795909	18.52818	0.640158	4.182485	3.570941	0.652178	23.28528	12.05936	1.224669	6.651809	0.444425
Native	0.091437	0.304522	21.19858	4.759762	1.911378	0.003401	21.50109	8.409533	0.782433	3.977332	18.39304	0.648554	4.417456	3.751195	0.565291	23.79617	12.88496	0.995259	4.972772	0.448902
Romance	0.136948	0.349345	23.4812	4.647805	2.083458	0.007157	23.47583	7.943248	0.666374	4.0211	18.07297	0.59202	4.305712	3.543449	0.657172	22.99821	12.55986	1.048799	6.176518	0.472759
Platform																				
Reddit	0.10953	0.407276	24.82821	4.693067	2.074249	0.008856	23.4199	8.059469	0.681728	3.925476	18.18832	0.637283	4.310731	3.548133	0.636779	23.31548	12.47371	1.095118	5.98496	0.464842
Origin																				
Native	0.091437	0.304522	21.19858	4.759762	1.911378	0.003401	21.50109	8.409533	0.782433	3.977332	18.39304	0.648554	4.417456	3.751195	0.565291	23.79617	12.88496	0.995259	4.972772	0.448902
NonNative	0.116361	0.446068	26.19847	4.667888	2.135736	0.010915	24.14429	7.927312	0.643709	3.905899	18.11103	0.633028	4.27044	3.471474	0.663767	23.13401	12.31845	1.132816	6.367083	0.470859

Table 29: Average feature value for each class in European Reddit dataset

Language	elongated	caps	textLength	sentenceWordLength	spellDelta	#	,	A	D	!	N	P	O	R	&	^	V	$	G	T
Bulgarian	0.045637	0.442754	47.61329	5.002576	2.431934	0.003203	18.6746	7.734081	0.661955	2.127078	19.81011	0.942773	3.941261	3.99032	0.697903	25.26162	15.09538	0.859557	3.853969	1.665673
Croatian	0.080865	0.478783	34.67014	4.275339	2.266001	0.004003	27.58377	6.905203	0.424477	3.680214	18.07147	1.24008	4.446077	3.103375	0.490654	23.41474	12.77189	1.226174	4.14963	1.465993
Czech	0.084067	0.635709	35.47558	4.443499	2.194824	0.016013	26.85237	6.735531	0.680383	3.314932	18.34356	0.848865	6.182438	3.097911	0.814335	21.39125	12.12287	1.014533	5.885559	1.248379
Dutch	0.043716	0.279903	23.15665	4.113568	2.071014	0.018215	30.91245	5.806777	0.559462	4.020326	17.84601	1.13334	5.201846	3.820706	0.931545	18.45049	12.46308	0.950243	4.464976	1.354825
English	0.085434	0.371349	21.83023	4.568756	1.901614	0.004202	21.96679	6.870478	0.66347	3.67349	19.98805	0.942354	5.412002	3.254661	0.505851	22.16638	13.70916	0.884653	5.002741	1.672278
Finnish	0.179503	0.506974	18.40206	4.64682	2.072471	0.015161	26.47725	6.161886	1.049091	2.838479	21.8915	0.705856	3.81716	3.343605	0.765183	20.8432	11.17465	1.000645	7.403091	1.307305
French	0.156421	0.175289	15.74376	4.23354	1.81647	0.009738	27.87867	6.570339	0.722683	4.696436	17.08998	0.932754	7.580605	3.163006	0.49988	18.78311	13.58684	0.737349	5.884548	1.380388
German	0.170109	1.071689	19.49514	4.265365	1.949051	0.013366	22.80533	6.096207	0.379153	3.802595	19.85505	0.628543	5.202074	3.637511	0.540804	21.32793	13.17592	2.843358	6.080004	1.048733
Italian	0.126743	0.629472	38.04427	4.375105	2.225569	0.011522	26.7525	5.95346	0.472983	3.11562	19.03714	1.052411	5.993354	2.836423	0.483773	21.72034	12.40848	1.340774	6.060826	1.389736
Lithuanian	0.064051	0.393114	16.13851	4.10015	1.881163	0.008006	29.80027	6.04475	0.217168	3.652228	19.08012	0.951206	5.975239	3.429494	0.740127	18.50385	12.52866	1.681679	7.552441	1.710356
Norwegian	0.088039	0.455981	31.05525	4.275941	2.15225	0.004857	23.56904	6.783094	0.477237	3.69968	19.10387	0.506321	4.858127	3.49034	0.702084	19.536	15.12712	1.158399	6.493702	1.546012
Polish	0.117694	0.615693	34.80865	4.446446	2.239269	0.004003	23.34802	6.922001	0.973077	3.59748	19.26496	1.062451	5.68915	3.928274	0.461022	22.00305	13.15328	0.893932	5.859354	1.094392
Portuguese	0.186173	0.331716	14.9436	4.083688	1.681252	0.004245	26.11844	6.097649	0.770062	4.147333	18.49615	0.76793	5.814337	3.261201	0.446971	19.49189	12.54189	1.059369	6.314097	1.471523
Romanian	0.092784	0.697392	37.59915	4.506189	2.383061	0.00849	24.34486	6.620464	0.618969	2.67316	18.77804	0.907115	5.630586	3.347164	0.685392	23.43081	12.6445	1.841557	6.556718	1.626775
Russian	0.028962	0.546259	23.26227	4.413232	2.081745	0.008045	27.44257	6.44434	0.573934	3.228758	19.80353	0.781066	5.274633	3.071143	0.417496	22.572	12.49638	1.088024	5.147817	1.060654
Serbian	0.238591	0.429944	65.84868	4.026885	2.417868	0.003203	26.38554	5.119752	0.228501	4.368395	18.50454	0.792557	7.622557	3.404931	0.734522	15.85563	16.37627	1.004581	4.15859	1.787661
Slovene	0.115292	0.508407	35.11369	4.407102	2.289986	0.01201	24.86683	6.192987	0.718101	3.031533	19.01094	0.858413	5.766331	3.591169	0.574849	19.7928	13.45947	1.072286	6.281117	1.583844
Spanish	0.124924	0.428745	32.45482	4.328968	1.917503	0.012735	20.6527	6.051386	0.483505	4.228833	19.28816	0.800446	6.647087	3.416105	0.523014	22.1384	14.68341	1.203385	5.739375	1.457004
Swedish	0.131068	0.573422	21.49393	4.362239	1.985631	0.005461	24.82713	6.288751	0.55395	4.392463	19.83012	0.842238	5.454503	3.306353	0.550951	20.25583	13.4127	1.004787	6.177564	1.480869
Family																				
Balto-Slavic	0.096936	0.506309	36.62437	4.389389	2.225435	0.00731	25.61815	6.512371	0.559691	3.375165	18.98566	0.934769	5.612414	3.452306	0.616483	21.09848	13.50113	1.105106	5.361188	1.452354
Germanic	0.122496	0.577516	22.7198	4.332874	2.046094	0.011412	25.71867	6.227351	0.603908	3.750559	19.70584	0.763272	4.906508	3.51962	0.698131	20.08274	13.07026	1.391168	6.12419	1.347592
Native	0.085434	0.371349	21.83023	4.568756	1.901614	0.004202	21.96679	6.870478	0.66347	3.67349	19.98805	0.942354	5.412002	3.254661	0.505851	22.16638	13.70916	0.884653	5.002741	1.672278
Romance	0.137395	0.452725	27.76587	4.30555	2.004908	0.009346	25.14745	6.258433	0.613561	3.771603	18.53895	0.892102	6.332285	3.20481	0.527826	21.11461	13.17272	1.23685	6.111278	1.465147
Platform																				
Reddit	0.1087	0.473284	27.42476	4.406853	2.046295	0.007845	24.53336	6.488775	0.610309	3.631362	19.32225	0.88846	5.56066	3.357394	0.584562	21.16543	13.38661	1.139065	5.604805	1.491874
Origin																				
Native	0.085434	0.371349	21.83023	4.568756	1.901614	0.004202	21.96679	6.870478	0.66347	3.67349	19.98805	0.942354	5.412002	3.254661	0.505851	22.16638	13.70916	0.884653	5.002741	1.672278
NonNative	0.117489	0.51179	29.53809	4.345694	2.100949	0.009221	25.50289	6.344587	0.590227	3.615448	19.07075	0.868101	5.616815	3.396201	0.614295	20.78733	13.26476	1.235169	5.832235	1.423727

Table 30: Average feature value for each class in non-European Reddit dataset

B. Figures

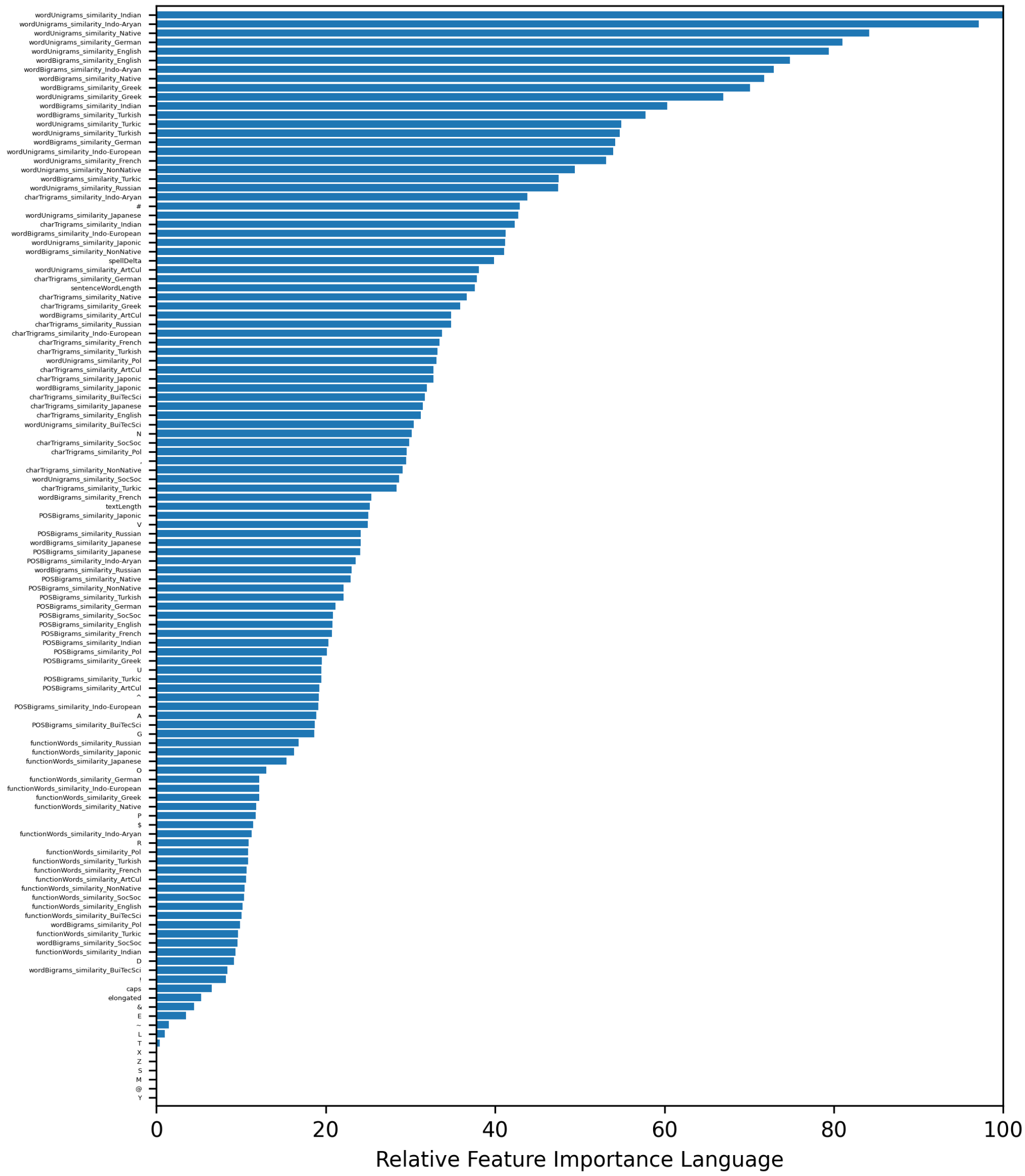

Figure 19: Twitter feature importance scores for **Language** from *Random Forest* classifier. Values are normalised to the highest score.

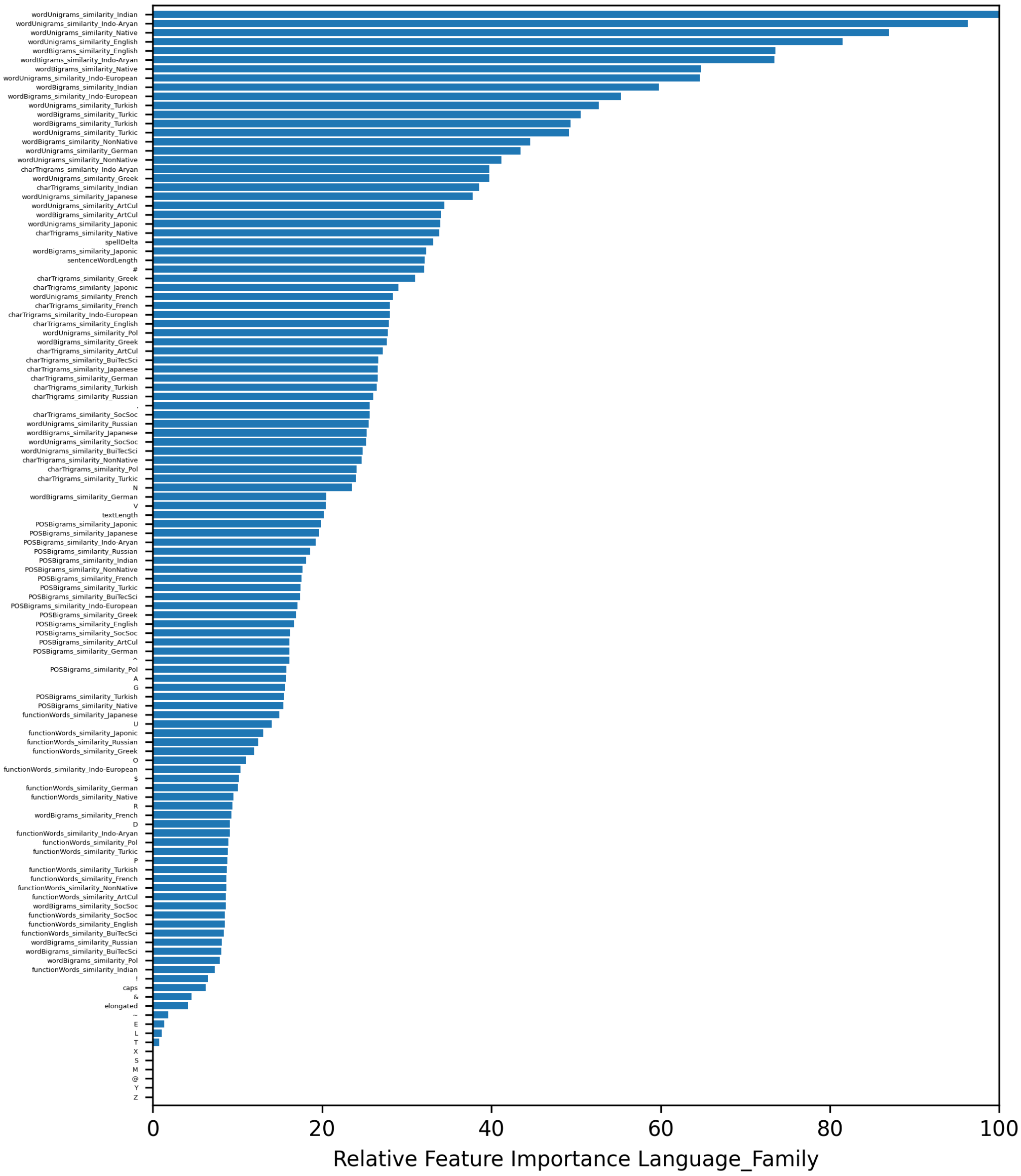

Figure 20: Twitter feature importance scores for **Language Family** from *Random Forest* classifier. Values are normalised to the highest score.

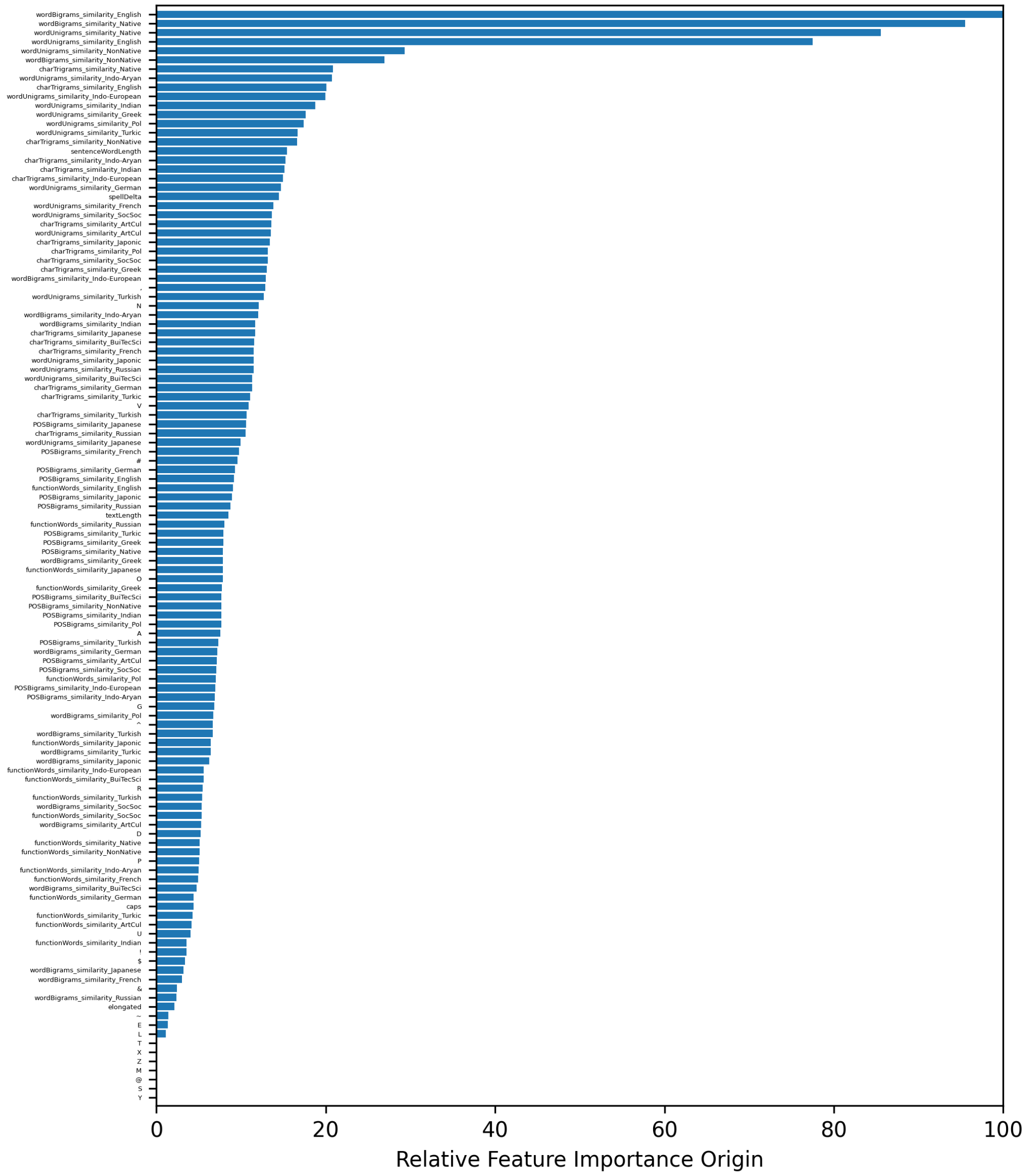

Figure 21: Twitter feature importance scores for **Origin** from *Random Forest* classifier. Values are normalised to the highest score.

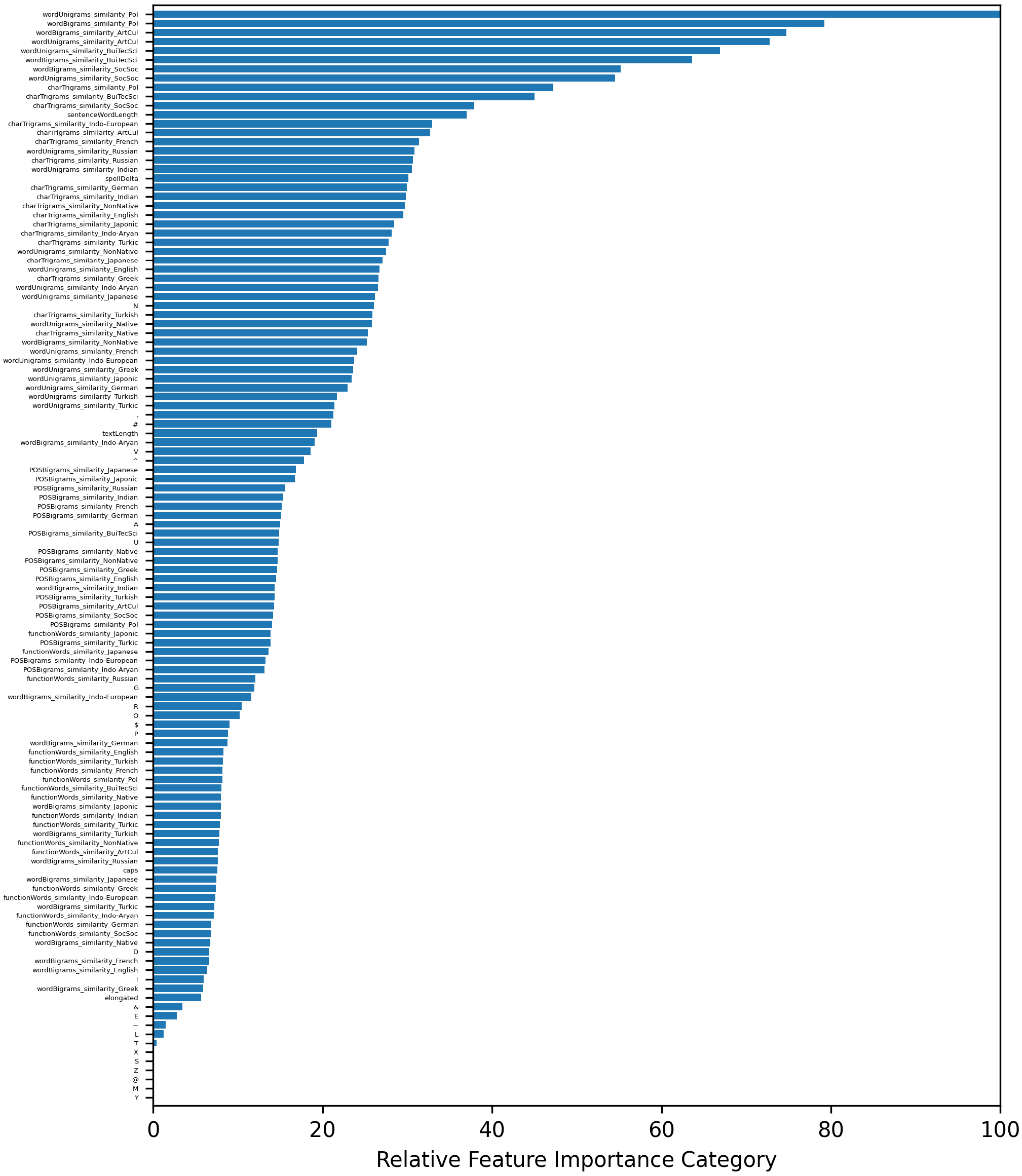

Figure 22: Twitter feature importance scores for **Category** from *Random Forest* classifier. Values are normalised to the highest score.

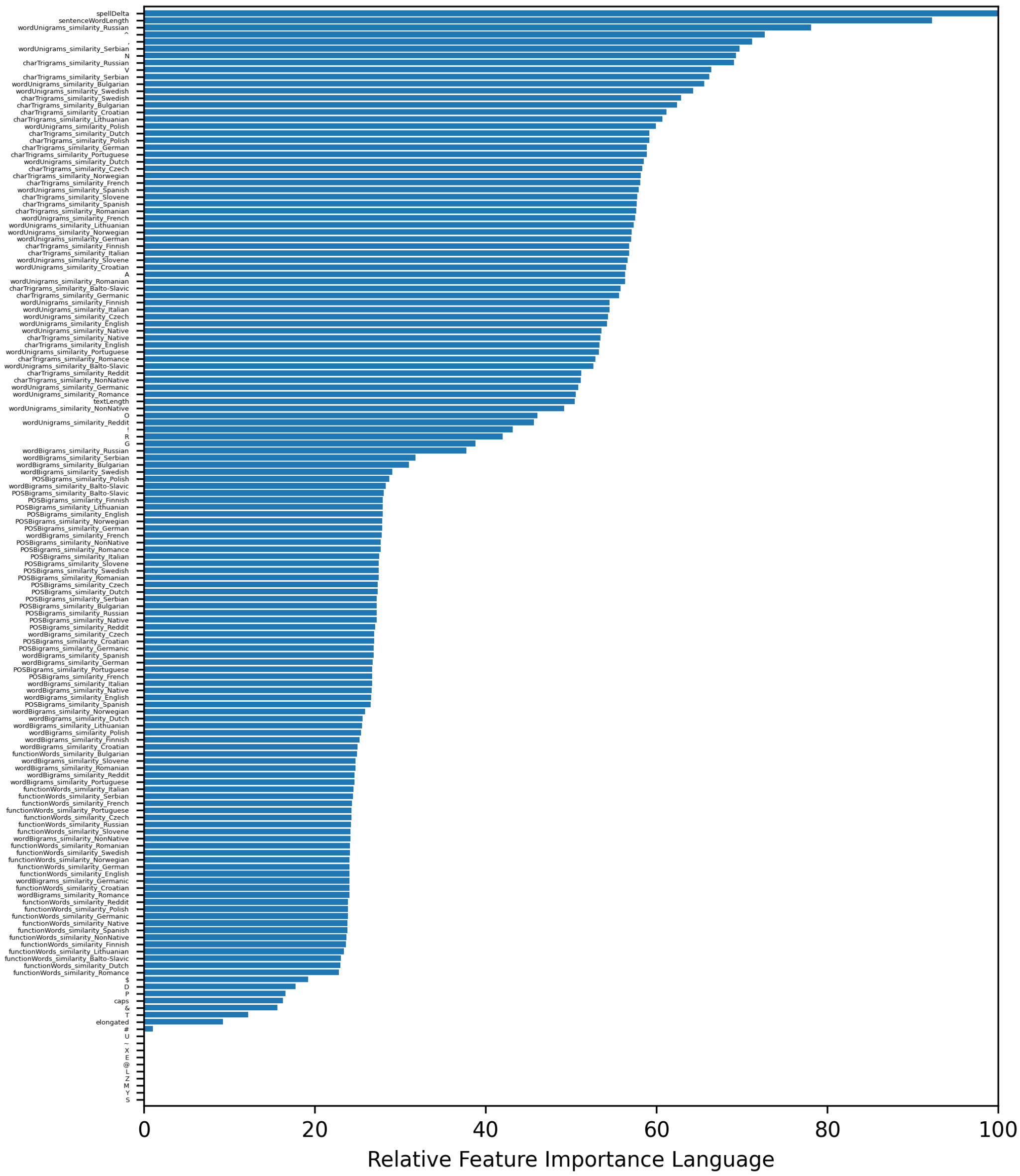

Figure 23: Reddit feature importance scores for **Language** from *Random Forest* classifier in European dataset. Values are normalised to the highest score.

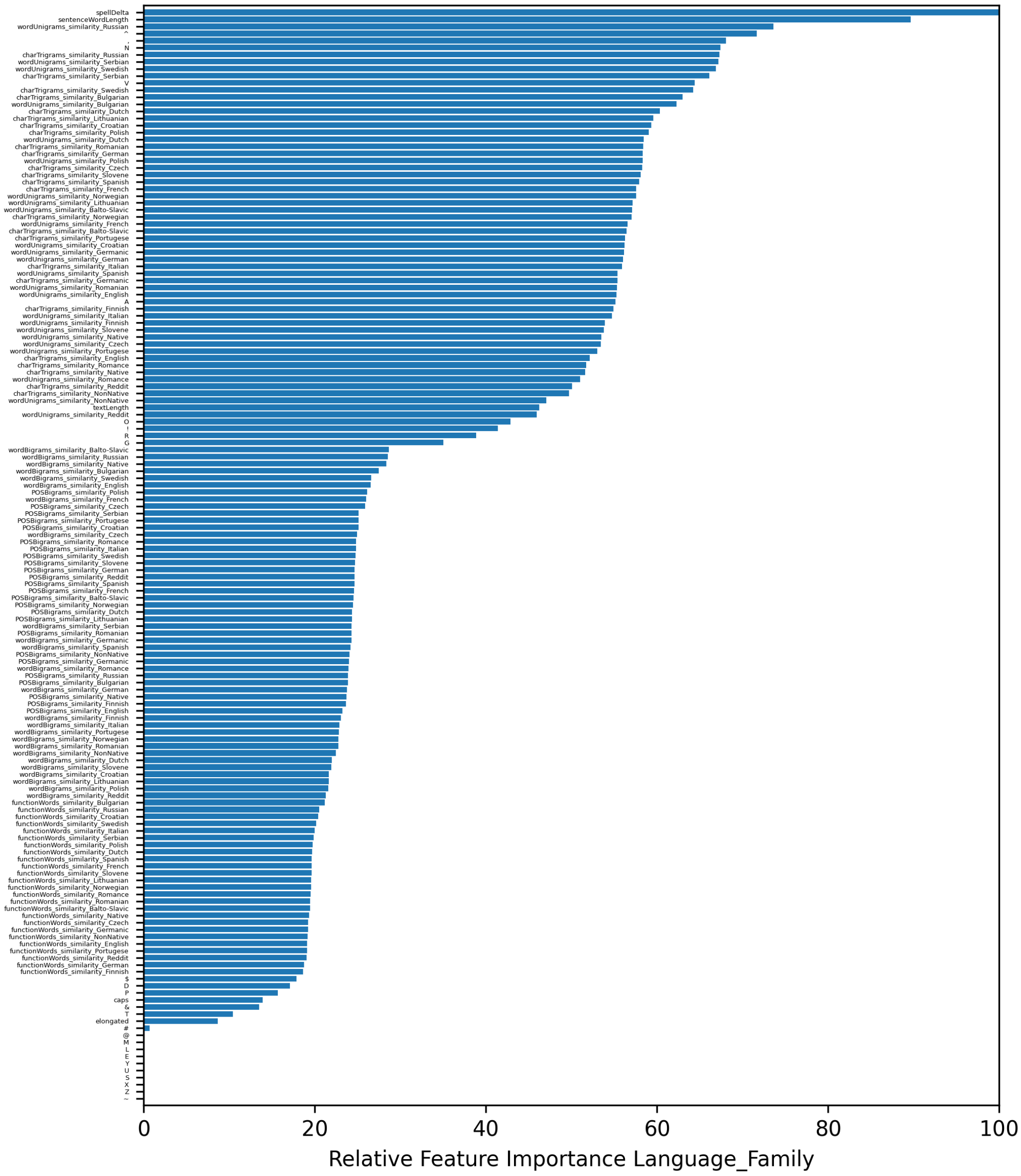

Figure 24: Reddit feature importance scores for **Language Family** from *Random Forest* classifier in European dataset. Values are normalised to the highest score.

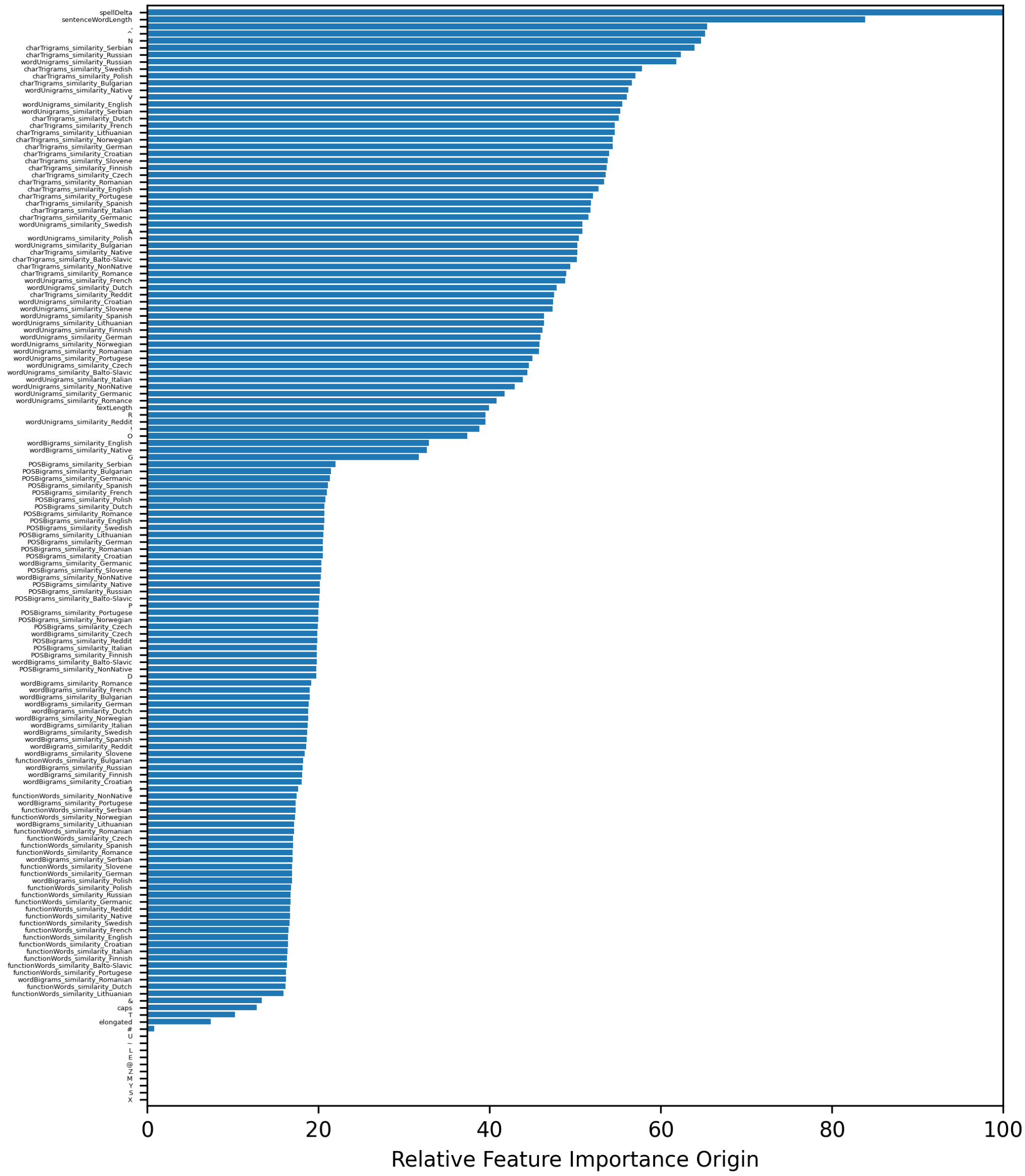

Figure 25: Reddit feature importance scores for **Origin** from *Random Forest* classifier in European dataset. Values are normalised to the highest score.

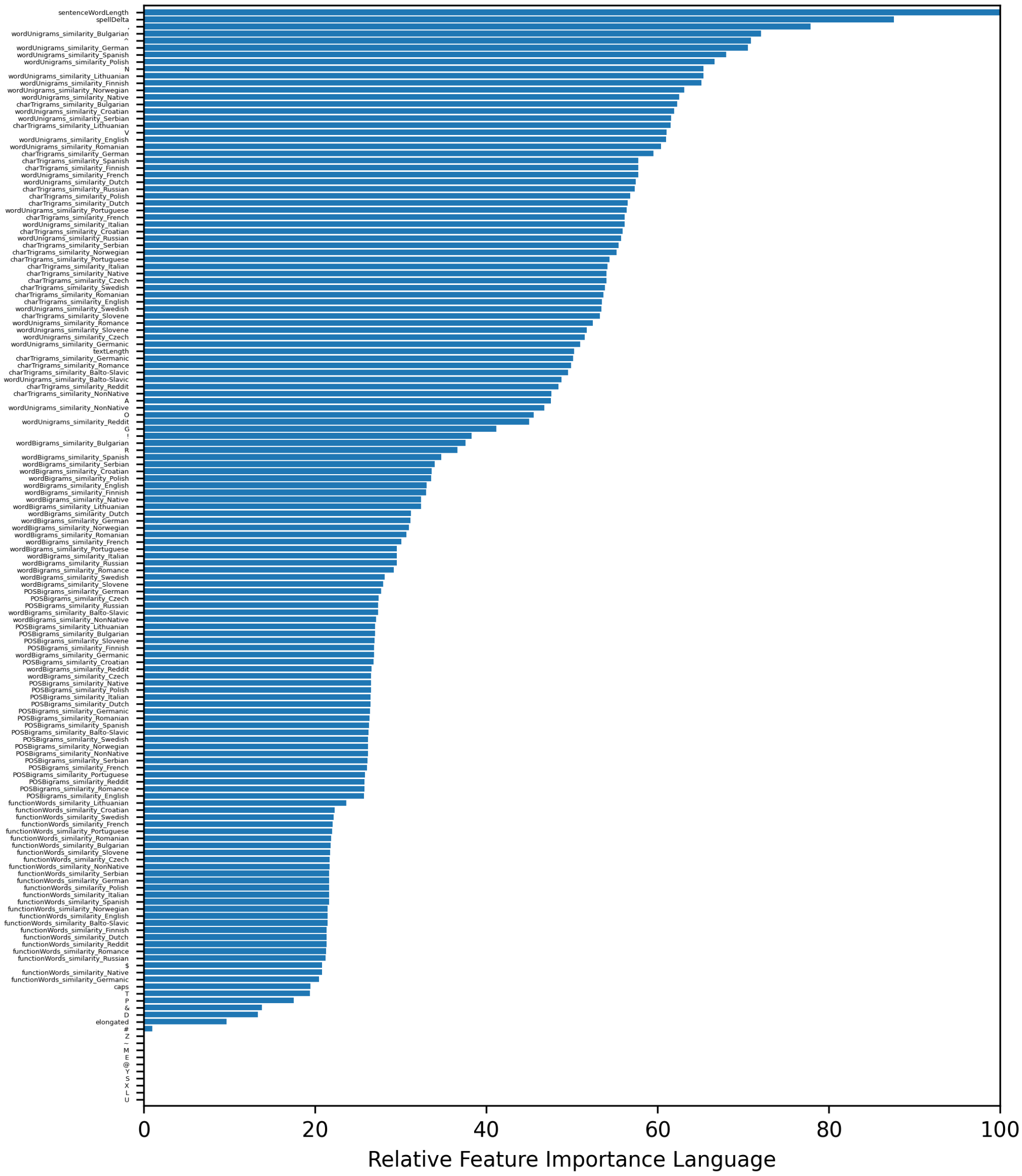

Figure 26: Reddit feature importance scores for **Language** from *Random Forest* classifier in non-European dataset. Values are normalized to the highest score.

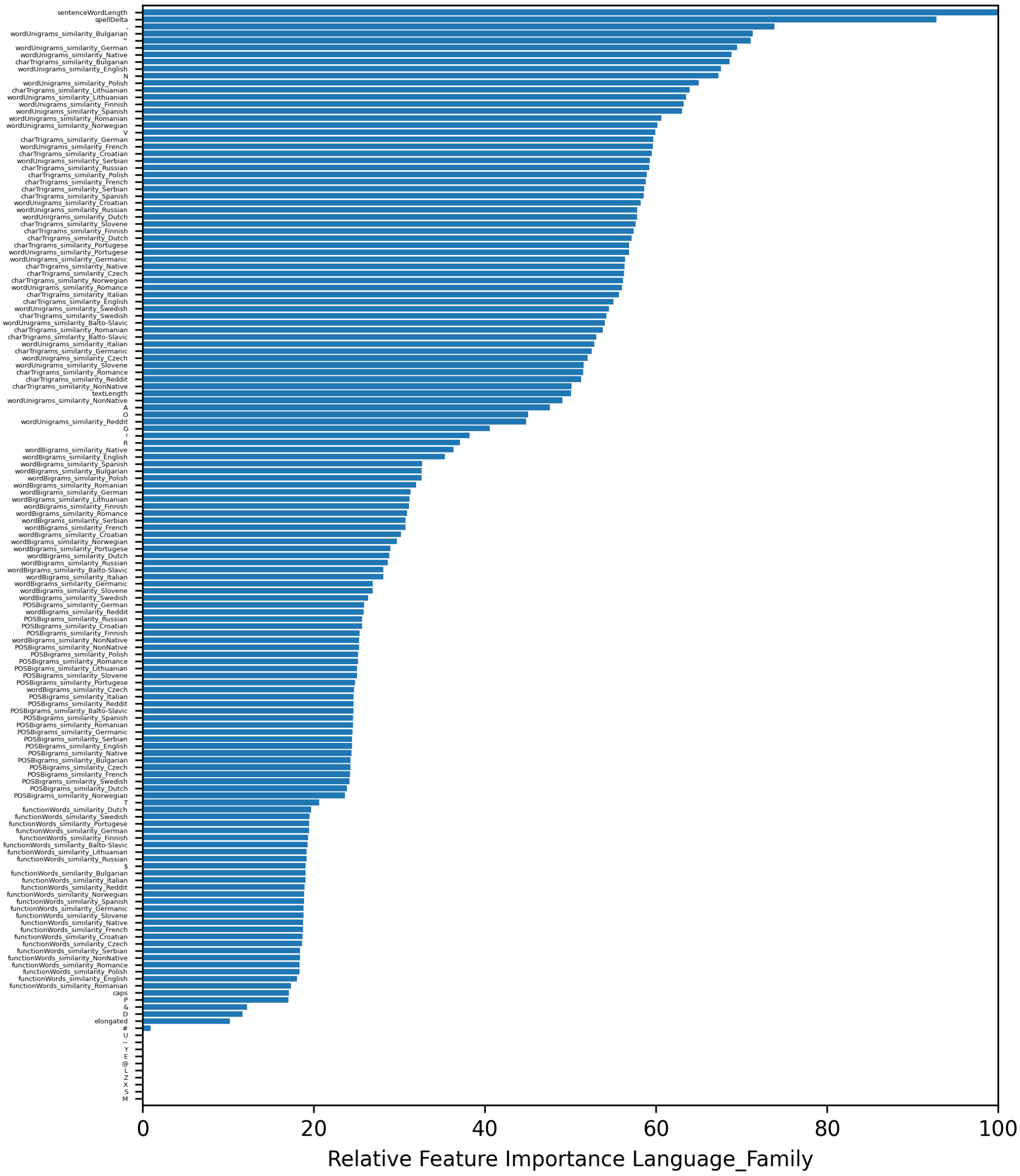

Figure 27: Reddit feature importance scores for **Language Family** from *Random Forest* classifier in non-European dataset. Values are normalised to the highest score.

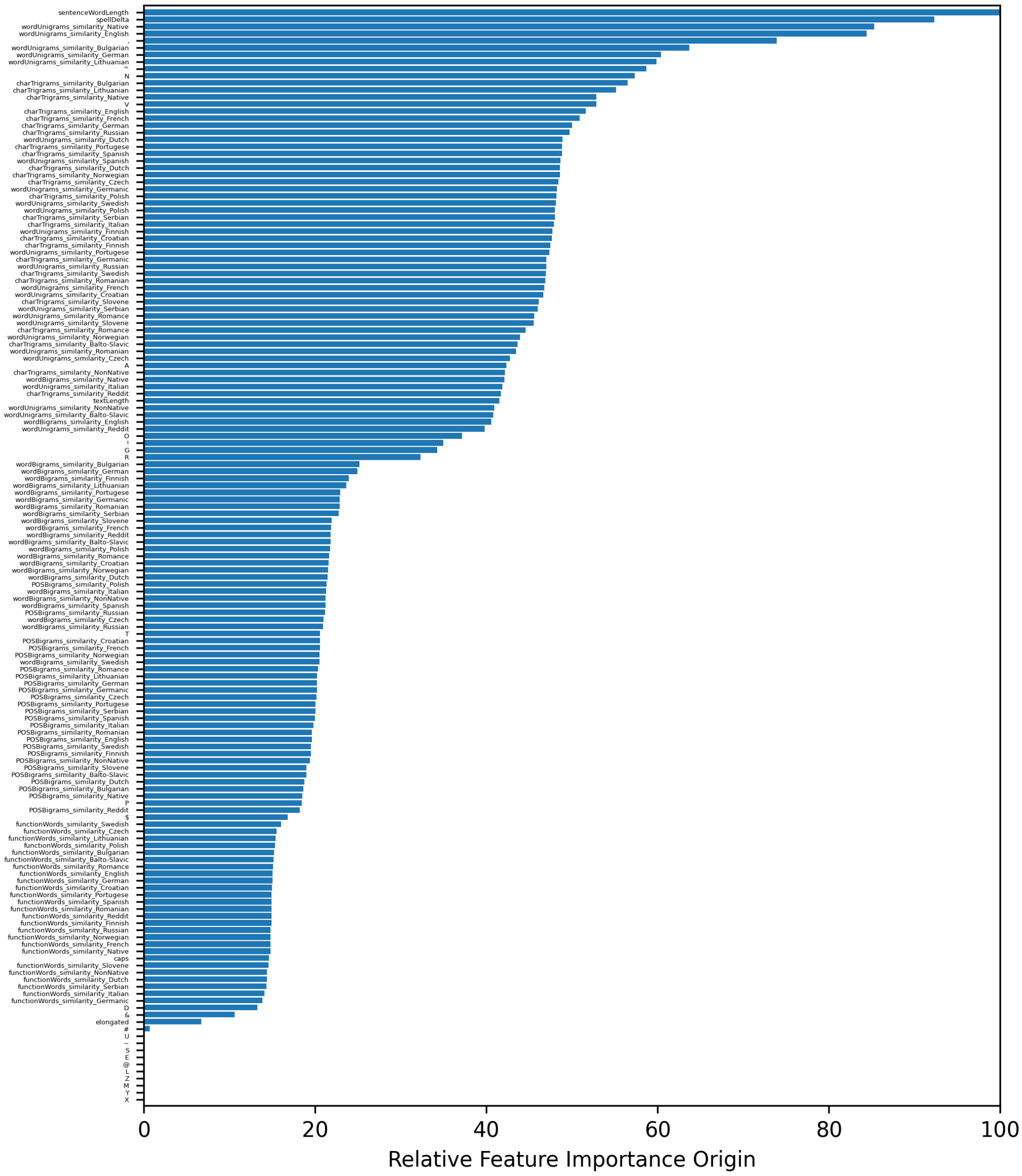

Figure 28: Reddit feature importance scores for **Origin** from *Random Forest* classifier in non-European dataset. Values are normalised to the highest score.

References

[Cano et al., 2014] Cano, A. E., Mazumdar, S., and Ciravegna, F. (2014). Social influence analysis in microblogging platforms - A topic-sensitive based approach. *Semantic Web*, 5(5):357–372.

[Cardoso and Roy, 2016] Cardoso, P. M. D. and Roy, A. (2016). Language identification for social media: Short messages and transliteration. In Bourdeau, J., Hendler, J., Nkambou, R., Horrocks, I., and Zhao, B. Y., editors, *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 611–614. ACM.

[Chen et al., 2010] Chen, Y., Lee, S. Y. M., Li, S., and Huang, C. (2010). Emotion cause detection with linguistic constructions. In Huang, C. and Jurafsky, D., editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 179–187. Tsinghua University Press.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Eke et al., 2019] Eke, C. I., Norman, A. A., Shuib, L., and Nweke, H. F. (2019). A survey of user profiling: State-of-the-art, challenges, and solutions. *IEEE Access*, 7:144907–144924.

[Ghanem et al., 2019] Ghanem, B., Buscaldi, D., and Rosso, P. (2019). Textrolls: Identifying russian trolls on twitter from a textual perspective. *CoRR*, abs/1910.01340.

[Goldin et al., 2018] Goldin, G., Rabinovich, E., and Wintner, S. (2018). Native language identification with user generated content. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3591–3601. Association for Computational Linguistics.

[He et al., 2019] He, X., Zhao, K., and Chu, X. (2019). Automl: A survey of the state-of-the-art. *CoRR*, abs/1908.00709.

[Kim et al., 2012] Kim, S., Bak, J., and Oh, A. H. (2012). Do you feel what I feel? social aspects of emotions in twitter conversations. In Breslin, J. G., Ellison, N. B., Shanahan, J. G., and Tufekci, Z., editors, *Proceedings of the Sixth International Conference on Weblogs and Social Media, Dublin, Ireland, June 4-7, 2012*. The AAAI Press.

[Lui and Baldwin, 2012] Lui, M. and Baldwin, T. (2012). langid.py: An off-the-shelf language identification tool. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 25–30. The Association for Computer Linguistics.

[Nguyen et al., 2013] Nguyen, D., Gravel, R., Trieschnigg, D., and Meder, T. (2013). "how old do you think I am?" A study of language and age in twitter. In Kiciman, E., Ellison, N. B., Hogan, B., Resnick, P., and Soboroff, I., editors, *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013*. The AAAI Press.

[Olson et al., 2016] Olson, R. S., Bartley, N., Urbanowicz, R. J., and Moore, J. H. (2016). Evaluation of a tree-based pipeline optimization tool for automating data science. *CoRR*, abs/1603.06212.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

[Qi et al., 2020] Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (2020). Stanza: A python natural language processing toolkit for many human languages. In Çelikyilmaz, A. and Wen, T., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 101–108. Association for Computational Linguistics.

[Rabinovich et al., 2018] Rabinovich, E., Tsvetkov, Y., and Wintner, S. (2018). Native language cognate effects on second language lexical choice. *Trans. Assoc. Comput. Linguistics*, 6:329–342.

[Schwartz et al., 2013] Schwartz, H. A., Eichstaedt, J. C., Dziurzynski, L., Kern, M. L., Blanco, E., Kosinski, M., Stillwell, D., Seligman, M. E. P., and Ungar, L. H. (2013). Toward personality insights from language exploration in social media. In *Analyzing Microtext, Papers from the 2013 AAAI Spring Symposium, Palo Alto, California, USA, March 25-27, 2013*, volume SS-13-01 of *AAAI Technical Report*. AAAI.

[Sun et al., 2016] Sun, J., Kunegis, J., and Staab, S. (2016). Predicting user roles in social networks using transfer learning with feature transformation. *CoRR*, abs/1611.02941.

[Volkova et al., 2018] Volkova, S., Ranshous, S., and Phillips, L. (2018). Predicting foreign language usage from english-only social media posts. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 608–614. Association for Computational Linguistics.

[Yao et al., 2018] Yao, Q., Wang, M., Escalante, H. J., Guyon, I., Hu, Y., Li, Y., Tu, W., Yang, Q., and Yu, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. *CoRR*, abs/1810.13306.