Automatic Detection of Idiosyncratic Phrases as Features for Authorship Attribution

Master's Thesis

Author: Lorenz Leitner, BSc
Supervisor: Ass.Prof. Dipl.-Ing. Dr.techn. Roman Kern

October 15th, 2020

Hypothesis

Hypothesis

People use different words and phrases according to their personalities.
$\Longrightarrow$ Authorship of texts can be ascertained based on these phrases.

Examples

Type	Examples
Unconscious	“let me tell you”, “that being said”, “I suppose”
Regional	“hella”, “neither here nor there”, “I reckon”
Internet	“u” instead of you, “iirc”, “afaik”
Errors	“could of”, “should of”, “would of”, “could care less”

Use Cases

Application constraints:

- Unstructured texts on the WWW (writing styles differ more)
- Either balanced topics or only one topic

Use cases:

- Phrase extraction
- Authorship attribution
- Forensic applications:
  - Anonymous threats
  - Hate speech

Choice of Data Set

Source of data: Reddit[1]

- Online discussion platform with informal text
- Data labeled by author and topic
- Topic = *Subreddit* (Sub-forums on Reddit limited to a specific topic)
  E.g. /r/gaming

The Reddit logo

[1] https://reddit.com

Background - Phrase Extraction

Phrase extraction in general:

- Used most often for *key* phrase extraction
- To summarize texts, create searchable terms, etc.
- Or to categorize texts by topic
- Can be done in general via linguistic features or pattern mining

Background - Phrase Extraction

Phrase extraction **here**:

- Extract *topic-agnostic* phrases
- To identify authors
- Only possible with specific input texts:
  One author and multiple topics, or multiple authors and one topic

Background - Authorship Attribution

All methods have in common:

- Training corpus (Labeled texts of known authorship)
- Testing corpus (Texts of unknown authorship)

Differences:

- What features they use, how they classify
- How they split up or combine author texts

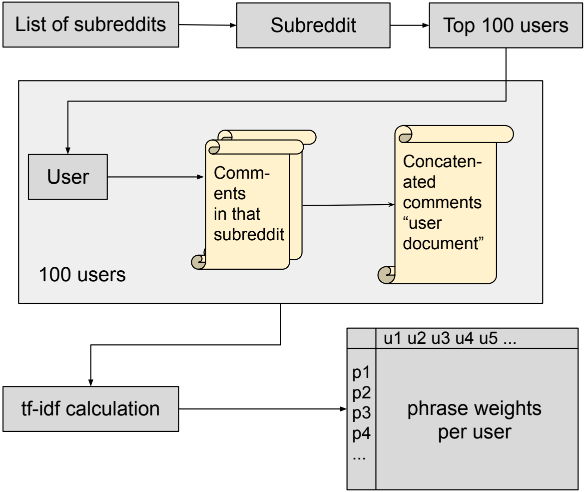
Concepts - Phrase Extraction

Phrase extraction - two methods:

1. ***n-gram* tf-idf** (“Method 1 tf-idf”)
   Works for multiple authors and one topic
2. **Sequential pattern mining** (“Method 2 seqpat”)
   Works for one author and multiple topics

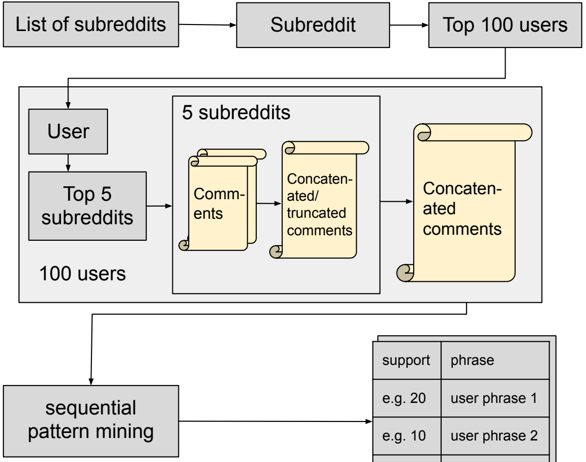
Concepts

Method 1 tf-idf
Phrase extraction



Concepts

Method 2 seqpat
Phrase extraction



support	phrase
e.g. 20	user phrase 1
e.g. 10	user phrase 2
e.g. 5	user phrase 3
...	...

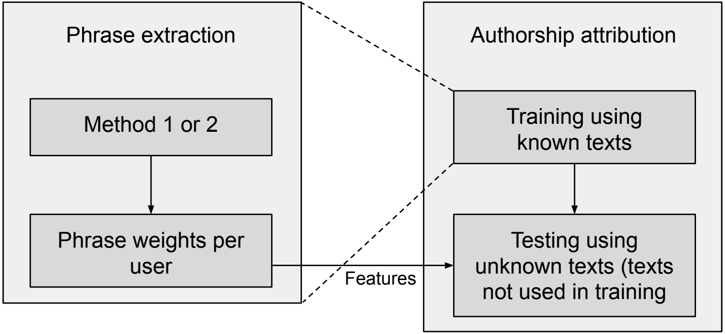
Concepts - Authorship Attribution

Authorship attribution:

- Training phase $=$ Phrase extraction, weighted phrases are features
- Testing phase: Unused texts of phrase extraction serve as “unknown” texts
- Attribution: Author with most *similar* phrases to phrases in unknown text is the most likely true author

Concepts

Authorship attribution



Concepts - Author Candidate Prediction

Attribution candidate author ranking:

- *Score* function: Score of candidate author $=$ n-gram counts in unknown text multiplied by weight of phrases in that author's dictionary

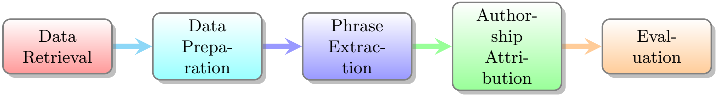
Score function

$$score = \sum |ngram| \times phraseweight$$

$$(\forall\, ngrams \cap phrases)$$

Implementation - Process

Pipeline



Implementation

General implementation aspects:

- Everything in Python on Arch Linux ($\Longrightarrow$ newest versions)
- Data retrieval: Pushshift.io API[2]
- Data cleaning: Redditcleaner[3]

[2] https://github.com/dmarx/psaw
[3] https://github.com/LoLei/redditcleaner

Implementation

General implementation aspects:

- Phrase extraction: sklearn's[4] TfidfVectorizer and spmf-py[5] for sequential pattern mining (based on SPMF [Fou+16])
- Attribution: sklearn's classification report for accuracy evaluation

[4] https://scikit-learn.org
[5] https://github.com/LoLei/spmf-py

Methodology

- Each user from a subreddit acts as the unknown author
- Attribution/comparison with the ≤ 100 users of the same subreddit
- Accuracy per subreddit: How many correct author predictions

Parameters

Method 1 tf-idf:

- Full tf-idf matrix (raw)
- Full tf-idf matrix (no stop word phrases)
- Top phrase dictionary for each user
- Unused texts from subreddit or from other subreddits

$$\implies \sum configurations = 6$$

Parameters

Method 2 seqpat:

- Phrase input type - Raw seqpat output or top phrase dictionary
- Algorithm - TKS or Gap-Bide
- Normalization method - L1 or min max

$$\implies \sum configurations = 8$$

Data Set - Subreddits

Either topic-specific or more general discussion

- AmItheAsshole
- askreddit
- books
- boxoffice
- classicwow
- games
- gaming
- HomeworkHelp
- MakeNewFriends
- movies
- nextfuckinglevel
- tifu
- todayilearned
- unpopularopinion
- worldnews

Data Set - Retrieval Strategy

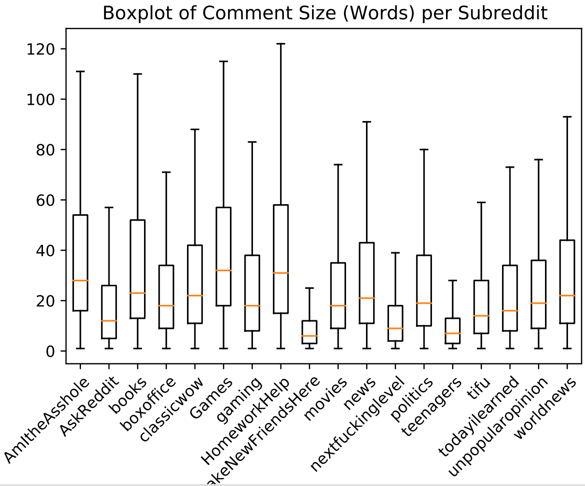
For each subreddit of the initial list:

- The top 100 most prolific users of the past 6 months are retrieved
- For these their last 10,000 comments in that subreddit are gathered
- Also the same for 5 other top subreddits per user

Data Set - Size

Number of subreddits	18
Number of subreddits after invalidation	15
Number of authors	1,748
Number of comments in the data set	10,642,641
Average comments per author	6,088
Number of comments in subreddit list	5,796,106

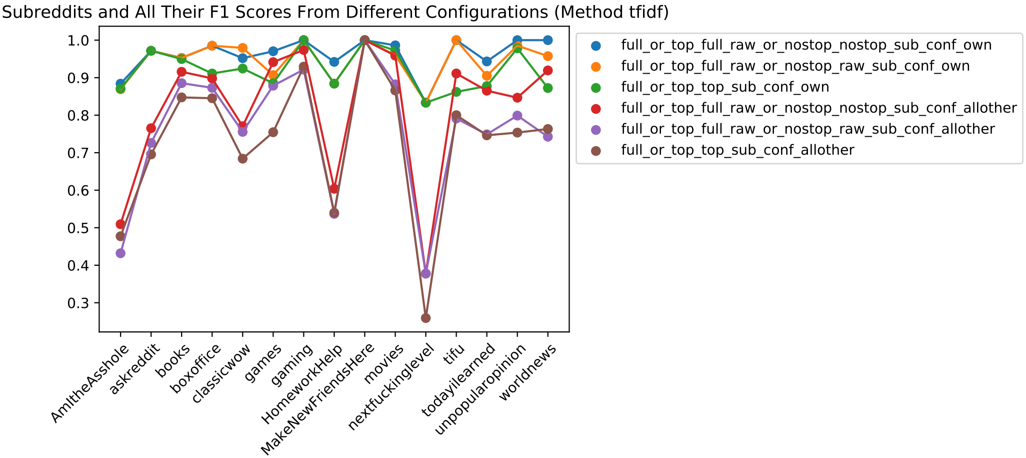
Data Set - Comment Sizes



Results - Method 1 tf-idf (F_1-scores)

Configuration parameters			**Results**	
Full/ Top dictionary	**Raw/ No stopword**	**Same/Other subreddits**	**Mean**	**Std Dev**
Full	No stopword	Same	0.961360	0.046247
Full	Raw	Same	0.946004	0.051400
Top		Same	0.919521	0.053162
Full	No stopword	Other	0.817124	0.177068
Full	Raw	Other	0.756692	0.172909
Top		Other	0.730771	0.180843

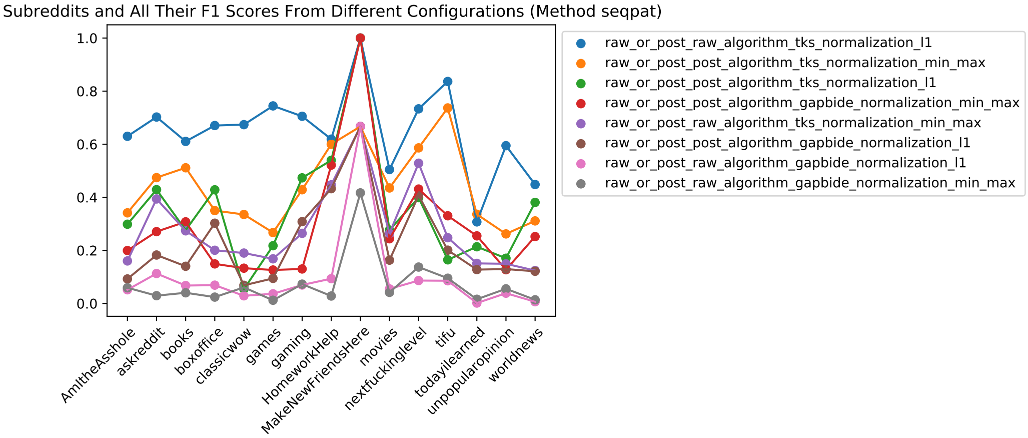
Results - Method 1 tf-idf (F_1-scores)



Results - Method 2 seqpat (F_1-scores)

Configuration parameters			**Results**	
Raw/ Post-processed (Top dictionary)	**Algorithm**	**Normalization**	**Mean**	**Std Dev**
Raw	TKS	L1	0.651988	0.156021
Post	TKS	Min Max	0.442776	0.143869
Post	TKS	L1	0.354701	0.214940
Post	Gap-BIDE	Min Max	0.298410	0.218361
Raw	TKS	Min Max	0.281912	0.153379
Post	Gap-BIDE	L1	0.229138	0.160407
Raw	Gap-BIDE	L1	0.097961	0.154975
Raw	Gap-BIDE	Min Max	0.073465	0.097375

Results - Method 2 seqpat (F_1-scores)



Results - All

All classification reports can be downloaded.[6]

[6] https://lolei.github.io/msc-reports

Discussion

- Method 1 tf-idf $>$ Method 2 seqpat
- Method 1 tf-idf: Better attribution within the same topic, worse outside of topic of phrase extraction
- Method 2 seqpat: May also be the reason why this method fares worse

Discussion - State of the Art F_1-scores

Model	Reddit		Average
Number of authors	10	50	
SVM+Stems [AG08]	35.1	21.2	60.0
SCAP [Fra+07]	46.5	30.3	65.3
Imposters [KSA11]	32.1	16.3	43.6
LDAH-S [SZB11]	43.0	14.2	49.9
CNN-char [RGB16]	58.8	37.2	73.4
M1 tf-idf		**96.1**	
M2 seqpat		**65.2**	

Discussion

- Comparison to state-of-the-art: Method 1 tf-idf outperforms on Reddit, but other models perform better on other domains [RGB16]
- Caveat: Both methods need specific topic constellations/labels in order to work at all
- Raw output works better for classification, top phrase dictionary is more convenient for human readers

Discussion - Sample Phrases

index	weight	longest phrases
0	0.03369	[do you mean nah]
2	0.027102	[need to]
3	0.026926	[sounds like]
4	0.026103	[it sound like]
6	0.025597	[you cant]
309	0.003178	[you have no reason to, you need to learn to, talk to her about it, i dont think its a, have the right to be, to be a part of]

Discussion - Sample Phrases

index	weight	longest phrases
0	0.015515	[may wanna]
1	0.014502	[you may wanna]
3	0.012109	[your opinion is]
4	0.011663	[..., your opinion is wrong, ...]
5	0.010649	[..., god i love, ...]
19	0.008196	[i loved that, ...]
22	0.007638	[oh man]

Discussion - Sample Phrases

index	weight	longest phrases
0	0.014816	[no ones saying]
4	0.012524	[imagine actually believing that, ...]

Discussion - Sample Phrases

index	weight	longest phrases
0	0.014602	[u are]
1	0.014365	[u can]
2	0.014212	[u have]
3	0.013986	[u will, u cant]
4	0.013378	[if u]

Discussion - Sample Phrases

index	weight	longest phrases
2	0.010935	[too many assholes, ...]
3	0.010025	[people are idiots, i wish that i, ...]
9	0.008088	[..., you can google it, ...]
12	0.007567	[you have a right to, ...]
14	0.007163	[..., im thinking about, think about my, obligated to, ...]
15	0.007108	[its impossible, ...]

Discussion - Sample Phrases

index	weight	longest phrases
17	0.006832	[doesnt mean anything, a couple of hours, maybe you can, ...]
21	0.006553	[..., i wouldnt know, pisses me of, ...]
28	0.006002	[i realized that]
30	0.005987	[that you know, ...]

Discussion - Sample Phrases

index	support	longest phrases
15	22	[i think, ...]
22	15	[..., i mean, ...]
25	12	[i thought, ...]
26	11	[trying to, i guess, ...]
28	9	[..., at least, ...]
30	7	[..., feels like, ...]
31	6	[it feels like, i dont know, instead of, ...]
32	5	[..., thought it was, ...]
33	4	[..., looking forward, ...]

Discussion - Sample Phrases

index	support	longest phrases
22	13	[..., i read, ...]
23	12	[a little, ..., nah]
24	11	[i thought, at least, ...]
25	10	[..., i mean, ...]
27	8	[..., trying to, couple of, ...]
28	7	[i dont know, like this, a couple, kind of, ...]

Discussion - Sample Phrases

index	support	longest phrases
29	6	[a couple of, i want to, i guess, i hear, i wish, yeah i, gotta, ...]
30	5	[..., i thought it, i disagree, so many, ...]
31	4	[your opinion is wrong, i feel like, feels like, ...]
32	3	[that being said, i thought it was, fuck fuck fuck, dont know if, pretty sure, i wonder if, ...]

Reflections

- Results confirm hypothesis
- Proposed methods only work with specific type of data
- Method 1 tf-idf works better than Method 2 seqpat
- Choice for method depends on input data
- This type of feature can now be used as a viable option or in addition to other features for authorship attribution

Future Work

- Advanced attribution methods
  Instead of “simple” *score* function

- More phrase extraction changes and implications
  What happens when more or less phrases are used?

- Traditional data sets
  Application of this method on traditional data sets, if labeled (balanced) topics are possible

Future Work

- Data set possibilities
  Full data set for download[7]

- Application in topic classification
  With different topic constellations, Method 1 tf-idf could be applied for topic classification

- Subreddit differences
  Why do some subreddits fare better or worse than others?

[7] https://lolei.github.io/msc-dataset

Bibliography I

[AG08] Ben Allison and Louise Guthrie. **Authorship Attribution of E-mail: Comparing Classifiers over a New Corpus for Evaluation.** LREC. Jan. 2008.

[Arg+07] Shlomo Argamon et al. **Stylistic Text Classification using Functional Lexical Features.** *Journal of the American Society for Information Science and Technology* 58.6 (Feb. 26, 2007), pp. 802–822. DOI: 10.1002/asi.20553.

[CH07] Jonathan H Clark and Charles J Hannon. **A Classifier System for Author Recognition Using Synonym-Based Features.** Mexican International Conference on Artificial Intelligence. Springer. 2007, pp. 839–849. DOI: 10.1007/978-3-540-76631-5_80. URL: http://www.cs.cmu.edu/afs/cs/Web/People/jhclark/pubs/MICAI07.pdf.

[De +01] Olivier De Vel et al. **Mining E-mail Content for Author Identification Forensics.** *ACM Sigmod Record* 30.4 (2001), pp. 55–64. DOI: 10.1145/604264.604272.

Bibliography II

[Fou+16] Philippe Fournier-Viger et al. **The SPMF Open-Source Data Mining Library Version 2.** Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer. 2016, pp. 36–40. DOI: 10.1007/978-3-319-46131-1_8. URL: http://www.philippe-fournier-viger.com/2016_PKDD_SPMF_VERSION2.pdf.

[Fou+17] Philippe Fournier-Viger et al. **A Survey of Sequential Pattern Mining.** *Data Science and Pattern Recognition* 1.1 (2017), pp. 54–77.

[Fra+07] Georgia Frantzeskou et al. **Identifying Authorship by Byte-Level N-Grams: The Source Code Author Profile (SCAP) Method.** *International Journal of Digital Evidence* 6.1 (Jan. 2007), pp. 1–18.

[Gri07] Jack Grieve. **Quantitative Authorship Attribution: An Evaluation of Techniques.** *Literary and Linguistic Computing* 22.3 (July 26, 2007), pp. 251–270. DOI: 10.1093/llc/fqm020.

Bibliography III

[KE07] Jussi Karlgren and Gunnar Eriksson. **Authors, Genre, and Linguistic Convention.** Proceedings from the SIGIR Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection. 2007.

[Ker13] Roman Kern. **Grammar Checker Features for Author Identification and Author Profiling.** CLEF 2013 Evaluation Labs and Workshop–Working Notes Papers. Citeseer, 2013. DOI: 10.1.1.666.9989.

[KSA11] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. **Authorship Attribution in the Wild.** *Language Resources and Evaluation* 45.1 (Mar. 2011), pp. 83–94. DOI: 10.1007/s10579-009-9111-2.

[OG16] Rebekah Overdorf and Rachel Greenstadt. **Blogs, Twitter Feeds, and Reddit Comments: Cross-Domain Authorship Attribution.** *Proceedings on Privacy Enhancing Technologies* 2016.3 (May 6, 2016), pp. 155–171. DOI: 10.1515/popets-2016-0021.

Bibliography IV

[RGB16] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. **Character-Level and Multi-Channel Convolutional Neural Networks for Large-Scale Authorship Attribution.** *arXiv preprint arXiv:1609.06686* (Sept. 21, 2016).

[Sum+20] Chanchal Suman et al. **Emoji Helps! A Multi-Modal Siamese Architecture for Tweet User Verification.** *Cognitive Computation* (Mar. 2, 2020), pp. 1–16. DOI: 10.1007/s12559-020-09715-7.

[SZB11] Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. **Authorship Attribution with Latent Dirichlet Allocation.** Proceedings of the Fifteenth Conference on Computational Natural Language Learning. June 2011, pp. 181–189.

[Zhe+06] Rong Zheng et al. **A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques.** *Journal of the American Society for Information Science and Technology* 57.3 (Feb. 1, 2006), pp. 378–393. DOI: 10.1002/asi.20316.

Bibliography V

[ZZ07] Ying Zhao and Justin Zobel. **Searching with Style: Authorship Attribution in Classic Literature**. Proceedings of the Thirtieth Australasian Conference on Computer Science - Volume 62. Australian Computer Society, Inc. Jan. 2007, pp. 59–68. DOI: 10.5555/1273749.1273757.

Questions?

State of the Art - Research

Methodology for research:

- Google Scholar
- ACM, IEEE, SpringerLink
- References in papers

Search terms:

- Phrase extraction
- Authorship attribution
- Stylometry
- Idiosyncrasy
- Data/Pattern Mining

State of the Art - Literature Review

Literature results:

Overall	Relevant	Read in detail
108	66	6

State of the Art - Scientific Work

Existing scientific work (examples):

- Large Scale Authorship Attribution using CNNs [RGB16]
- Cross-Domain Authorship Attribution [OG16]
- Multi-Modal Content [Sum+20]
- Classification With Synonym-Based Features [CH07]
- Sequential Pattern Mining [Fou+17]

State of the Art - Features

Classification features used in other works:

- Character counts [Gri07]
- Writing errors [Ker13]
- Unique vocabulary [De +01]
- Part-of-speech tags [ZZ07]
- Sentence structure [KE07]
- Semantic features [Arg+07]
- Topic-based features [Zhe+06]
- Application-specific [Zhe+06]

Implementation - Score Function

Simplified *score* function:

```
calculate_score(candidate author ngrams weights,
                unknown text):
  score = 0.0
  for all ngrams in unknown text:
    score += (frequency of ngram in unknown text) ×
      (weight of ngram in candidate author weight list)
  return score
```