Fingerprints in SMS Messages: Automatic Recognition of a Short Message Sender Using Gradient Boosting

Branislava Šandrih
Faculty of Philology, University of Belgrade
branislava.sandrih@fil.bg.ac.rs

Abstract

This paper considers the following question: Is it possible to tell who is the short message sender just by analyzing a typing style of the sender, and not the meaning of the content itself? If possible, how reliable would the judgment be? Are we leaving some kind of "fingerprint" when we text, and can we tell something about others based just on their typing style? For this purpose, a corpus of $\sim$ 5,500 SMS messages was gathered from one person's cell phone and two gradient boost classifiers were built: first one is trying to distinguish whether the message was sent by this exact person (cell phone owner) or by someone else; second one was trained to distinguish between messages sent by some public service (e.g. parking service, bank reports etc.) and messages sent by humans. The performance of the classifiers was evaluated in the 5-fold cross-validation setting, resulting in 73.6% and 99.3% overall accuracy for the first and the second classifier, respectively.

1. Introduction

It does not happen so rarely that we just see a message and know the sender, without even looking at the message header. Even though we miss signature, voice, mimics, sound and so many other components that written and oral communication contains, just by usage of emoticons, abbreviations, specific typos, grammar misses or specific use of punctuation — we can assume who are we communicating with. This is primarily true for the people with specific typing style. In the case of very short message, e.g. "Where are you?", determination of the sender can become more difficult. The task is not easy at all even for humans, especially when we do not have any other information such as cell phone model of the sender, operative system the sender uses, location etc.

In this paper, Gradient Boost (Friedman, 2001; Hastie et al., 2009) model was trained in order to predict the message sender. This is done by using lexical and syntactic features. Extracted features are put on disposal as CSV file. The dataset, Python module for feature extraction and code for model training and evaluation are available at github.[1] Since the external validation dataset was not available, the performance estimation is done by using 5-fold cross validation (CV). Although no external language tools were used (such as dictionaries or taggers), the method is designed to achieve the best performance on Serbian text messages.

This paper is organized as follows. Related work done so far is listed and briefly described in section 2.. In Section 3. we describe underlying SMS dataset and in Section 4. we describe extracted features. In Section 5. the steps of creating classifiers and most relevant features used by these two models are described. Afterwards in Section 6. we display classification results of our models. Finally, we conclude paper and state future plans in Section 7.

[1] Github repositorium, https://github.com/Branislava/sms_fingerprint

Keywords: SMS, short messages, classification, features extraction, gradient boosting

2. Related Work

Regarding the problem of author recognition, two most prominent research fields are Authorship Attribution (AA) and User Profiling. Most of the work done so far was related to the semantic analysis of the content (Pennebaker and King, 1999; Mairesse et al., 2007). Concerning AA, another approach in solving task of automatic recognition of the given text's author is by observing *stylometric* cues. These stylometric features (Roffo et al., 2014: 33) include *lexical* (counts of words and characters in text) and *syntactic* (punctuation and emoticons) features. After extraction of these features, they are typically used with discriminative classifiers, so that each author represents one class. A survey about application of AA to Instant Messaging (IM) was conducted in (Stamatatos, 2009). In (Zheng et al., 2006) stylometric features were used with Support Vector Machine (SVM) and Artificial Neural Network (ANN) classifiers, while authors of (Abbasi et al., 2008) applied PCA projection for AA on corpora containing E-mails, IMs, feedback comments and even program code. Similar work on AA in IMs was also conducted in (Abbasi and Chen, 2008).

In (Orebaugh and Allnutt, 2009) authors identify participants within IM conversation by observing sentence structure and usage of special characters, emoticons and abbreviations. Writing style of individuals is in focus in (Roffo et al., 2014). Authors analyze whether special interactional behavior, as the one present in the live communication, can emerge in chats. They also inspect if certain personality traits affect writing style. Authors conclude that some traits significantly affect chatting style and that some of them can be very effective with identifying a person among diverse individuals.

Similar research is conducted in (Eckersley, 2010) and (Laperdrix et al., 2016). These authors are more oriented at determining how trackable certain computer configuration is, based on Web browser version, the underlying operating system, the way emojis are displayed within Web browser, etc.[2]

3. The Dataset

A corpus of 5551 short messages structured as XML was collected from one person's cell phone in a 4-years time period. Each message contains information about sender's phone number, a date the message was sent, content of the message and other technical information. The corpus mostly consists of messages in Serbian, typed in both letters, Latin and Cyrillic, with some messages in English and German. The following two messages from the corpus are written in different letters, asking the same question in two different ways.[3]

<sms address="+381643057***" date="1424530897293" type="1" contact_name="Gri***" readable_date="21.02.2015 4:01:37 PM" body="Disiiiiiiiiiiiiiiiiiiii :-)" />

<sms address="+381600854***" date="1511436828568" type="0" contact_name="Мал***" readable_date="23.11.2017. 12:33:48 PM" body="Где си?" />

Attribute *type* represents whether the message was sent (value 1) or received (value 0). DTD for this corpora and the total list of features and their values is available on github.[4]

4. Features and Fingerprinting

Stylometric features[5] were extracted only from *body* attribute of `<sms>` elements and they can be divided into two categories: 1) lexical features and 2) syntactic features. This categorization is obtained from (Roffo et al., 2014: 33). Bag-of-Words features were not added to the final model as it turned out

[2] Am I Unique?, `https://amiunique.org/`

[3] Both messages contain "Where are you?" question, which is a common greeting line in Serbian. First message contains informal dialect-specific greeting, what can be observed by use of repeated letters and an emoticon. Second message is written in Cyrillic, that is normally less used in informal communication.

[4] DTD and extracted features
`https://github.com/Branislava/sms_fingerprint/tree/master/dataset`.
For XML files with corresponding DTD, features can be extracted with Dataset class
`https://github.com/Branislava/sms_fingerprint/blob/master/features_extraction/dataset.py`

[5] Other authors use similar set of features, naming them "linguistic features", e.g. in (Ebert, 2017: 55) and (Repar and Pollak, 2017).

they did not improve classification accuracy in this case. Along with dominant stylometric features, the final feature set was enriched with an additional set of common abbreviations and slang words.

4.1. Lexical Features

In this category, 11 features were extracted: number of characters, number of Cyrillic characters, diacritics count, number of umlauts, number of uppercase characters, number of lowercase characters, digits count, number of alphabet characters, number of occurrences of same consecutive characters,[6] number of sentences starting with lowercase character[7] and number of words starting with "ne".[8]

These features were selected after careful analysis performed by human, as it seemed they could help with distinguishing message senders that make specific typos and grammatical mistakes, or the ones that write too long or very short messages. For example, the minority of senders write in uppercase or in Cyrillic only and ones that write in German (hence the umlauts count).

4.2. Syntactic Features

These features can be divided into two categories: 1) emoticons and 2) punctuation features.

4.2.1. Emoticons

Ninety-eight different emoticons were listed and classified into 9 groups. First group consists of emoticons that represent a smile (smiley), second one contains emoticons that have a happy face (happy) and similarly other groups are formed: sad, surprised, kiss, wink, tongue, skeptic, miscellaneous.[9] In this specific dataset not all emoticons are present, and therefore the ones that are missing were discarded during preparation phase, keeping thirty-four emoticons. They are represented with corresponding regular expressions:

kiss `:* :*{2,} :-* :-*{2,}`

tongue `:-p{2,} :p{2,} :-P{2,} :-P{2,}`

sad `:( :-({2,} :-( :({2,} :-'( :-'({2,}`

smiley `:-) ;) :) ({2,}: (:`

wink `;-) ;){2,} ;-){2,}`

happy `xD{2,} xD :D{2,} :-D{2,} :D`

skeptic `:/{2,} :/`

surprised `:o :-o`

kiss `=D =] 8-)`

An absolute count of each emoticon appearance per message was added as a single feature. Afterwards, additional nine features were added as aggregated count of each emoticon type (e.g. total number of smiley emoticons, total count of all happy emoticons in a message etc.).

Emoticons have been useful in many research topics, such as sentiment analysis (Read, 2005; Škorić, 2017) or in short messages interpreting (Walther and D'Addario, 2001; Derks et al., 2007).

[6] Repeated characters, like in word "Disiiiiiiiiiiiiiii" make an impression of a person in a good mood.

[7] If one starts most sentences with lowercase characters, that is probably due to mobile phone operating system, what can be a partially identifying feature.

[8] Negation of verbs in Serbian is made by adding word "ne" before the verb, separately. It is a common mistake that people type this as one word, e.g. instead of correct negation "ne mogu", one could write incorrectly "nemogu".

[9] All emoticons with corresponding regular expressions
`https://github.com/Branislava/sms_fingerprint/blob/master/features_extraction/emoji.py`

4.2.2. Punctuation

For this dataset, nine punctuation-related features were considered to be important for sender dissemination: count of exclamation marks, count of question marks, count of dots, count of commas, total count of present punctuation, times when space followed punctuation, number of sentences separated by dot that does not precede space, count of `..` (double dot) and count of `??` tokens.

These features are extracted with an idea that certain people always make similar typing mistakes. For example, some people tend to write "bad" punctuation, such as two dots (instead of one or three) or do not write spaces after punctuation, they "glue" the sentences together with a dot and no additional blank space, etc.

4.3. Combining Lexical and Punctuation Features

Sixteen more features were added as a ratio of already mentioned feature counts. These features are ratios of: number of exclamation marks/question marks/dots/commas/total punctuation/alphabetic characters/diacritics/umlauts/cyrillic/uppercase/lowercase/digits and message length, ratio of upper and lowercase characters and ratio of punctuation/cyrillic/digits and alphabetic characters. The list of all extracted text is available at github.

4.4. The Abbreviation Features

Abbreviations are very common in this specific dataset, and therefore a list of total 135 different abbreviations was made. Some of the abbreviations are: *ae* (hajde - "come on"), *dog* (dogovoreno – "deal"), *dop* (dopisivati – "chat"), *k* (ok – "ok"), *msm* (mislim – "I think"), *mzd* (možda – "perhaps"), *najvrv* (najverovatnije – "most probably"), *nmg* (ne mogu – "I cannot"), *nmvz* (nema veze – "nevermind"), *nnc* (nema na čemu – "you welcome"), *np* (nema problema – "no problem"), *npm* (nemam pojma – "I have no clue"), *nzm* (ne znam – "I don't know"), *stv* (stvarno – "really"), *ustv* (u stvari – "actually"), *vcs* (večeras – "tonight"), *zvrc* (zovi me –"call me") etc.

5. Classification Model and Results

Two experiments were run, both with binary classification task. After several different classifiers evaluation, Gradient Boost model (Friedman, 2001; Hastie et al., 2009) turned out to achieve the best precision and accuracy in both cases. Detailed classifiers comparison is given in Section 6.

5.1. First Experiment: Specific Person vs. Others

The classification model was built to tell whether an unseen message was written by the native cell phone owner (positive class, label 0) or by someone else (negative class, label 1). Class labels are induced from *type* attribute of `<sms>` element explained in Section 3. There are 2,170 instances belonging to positive class and 3,381 instances belonging to negative class, making this dataset slightly unbalanced.

List of fifteen features that had the strongest influence on the model can be seen in Figure 1.[10] The majority of the most influential features are lexical and punctuation features: ratio of uppercase characters and message length (significant for persons who write in uppercase), message length, ratio of upper and lowercase letters, presence of spaces after punctuation, usage of question marks and dots. The fact that these features showed up as most important was not a surprise, since it was expected that exactly these features are what makes person's typing style distinguishable from other senders'.

5.2. Second Experiment: Human vs. Machine

Although the dataset is quite unbalanced in this case, the task is much easier than the previous. There are 918 instances belonging to positive class (label 0, messages sent from public services such as bank reports, parking services, mobile service providers etc.) and 4,633 messages sent from humans (label 1). Top 15 features that had the strongest influence were shown in Figure 2.

[10]Order of these most important features may vary during different cross-validation folds (depending on the message instances selected for the training set).

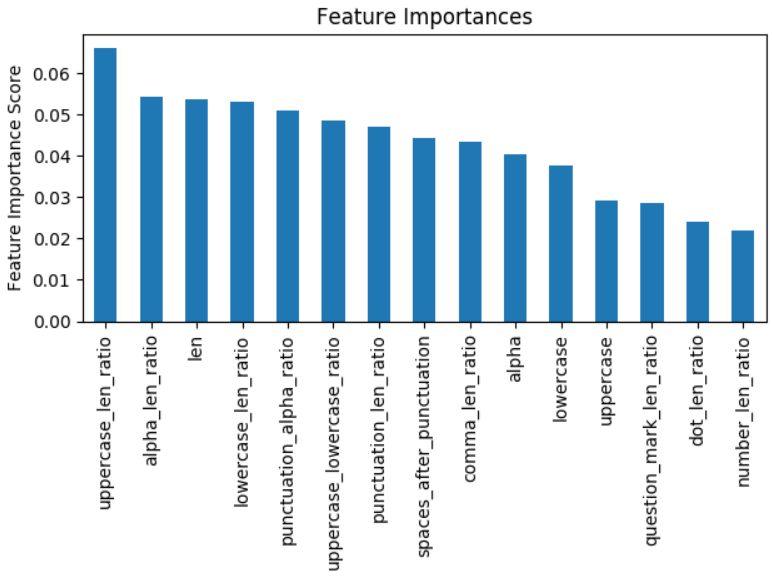

Figure 1: Most important features for the first model

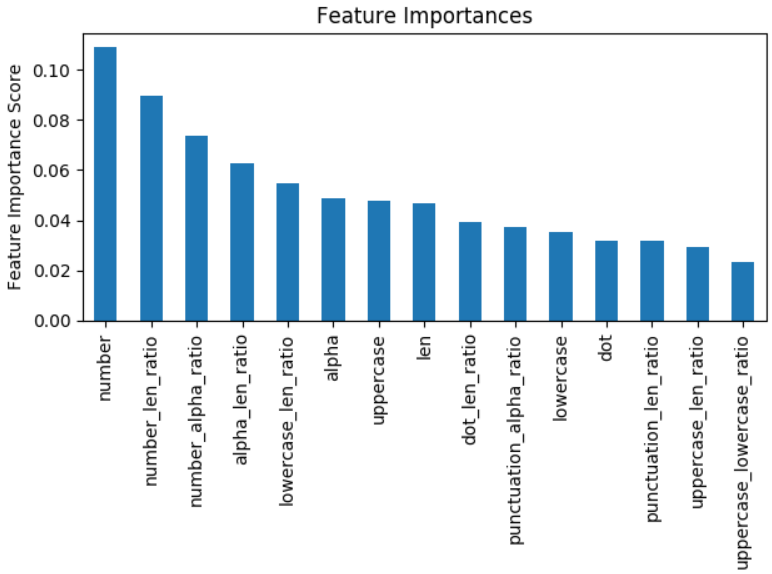

Figure 2: Most important features for the second model

Most of these features are related to presence of numbers, which is expected. These reports mainly consist of different digits that represent date and time when the report was sent, amount of money in a bank account, time when the parking card expires, etc. Similarly, these messages length is also somewhat specific, i.e. reports usually contain more tokens than regular humans' messages. Another common feature is number of the dot character used in comparison to other characters. Reports are usually longer and contain a few sentences, each concluded with a dot, which could not be guaranteed for informal messages. We can also notice that features have stronger influence (higher scores, y-axis) than in the previous experiment.

6. Results

We tested and compared the following algorithms implemented in *SciKit-Learn*, Machine Learning module for Python (Pedregosa et al., 2011):

SVM Support Vector Machine is a supervised machine learning algorithm that can be used for both classification and regression problems. Classification is performed by finding the hyper-plane that separates samples from different classes with the highest possible margin. In the case that samples are linearly separable, i.e. it is possible to find a hyper-plane that separates training samples good enough, SVM is linear. If samples are not linearly separable, a kernel function for the classifier should be selected. This means mapping all samples into other, higher-dimensional space, where the separating hyper-plane can be obtained. Beside kernel function, parameters of this classifier are: penalty parameter *C*, *gamma* (ignored if kernel is not Radial Basis Function (RBF), default value *auto*), *tol* (tolerance for stopping criterion, default value is 0.001), *class_weight* (if not given, all classes are supposed to have weight 1) and *max_iter* (maximum number of iterations; by default, this number is unlimited).[11]

MLP Artifical Neural Network (ANN) is a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks. Multi-layer Perceptron is a class of feed-forward ANNs that consists of at least three layers of nodes (input, hidden and output layer). It is a supervised learning algorithm that, given a set of features, can learn a non-linear function approximator for either classification or regression task. As for parameters, except regularization term *alpha* = 1, we used default values: *hidden_layer_sizes* = 100, the rectified linear unit function (relu) for *activation* parameter, stochastic gradient-based optimizer (adam) for *solver*, *tol* = 0.0001 as tolerance for optimization etc.

Gradient Boost Gradient boosting is a sequential technique that combines a set of weak learners, usually decision trees, and delivers improved prediction accuracy in an iterative fashion. Trees are added one at a time and a gradient descent procedure is used to minimize the loss when adding new trees. After calculating error or loss, the outcomes predicted correctly are given a lower weight and the ones miss-classified are weighted higher, until best instance weights are found. Before building the final classifier, grid search was performed in order to find optimal classifier parameters. At the end, model was tuned with next parameter values: *learning_rate* = 0.1, *n_estimators* = 160, *min_samples_split* = 10, *min_samples_leaf* = 30, *max_depth* = 9, *max_features* = 11, *subsample* = 0.8 and *random_state* = 10.

The performance of the classifiers was evaluated in the 5-fold CV setting using the following basic measures: accuracy, precision, recall and F-score. As a baseline, a classifier that always predicts the majority class in the dataset was used.

Detailed results for the 1^{st} experiment are given in Table 1.

Classifier	**Accuracy**	**Precision (+ class)**	**Recall (+ class)**	**F-score (+ class)**	**Precision (- class)**	**Recall (- class)**	**F-score (- class)**
Baseline	0.609	0.000	0.000	0.000	0.609	1.000	0.757
Linear SVM (C=0.025)	0.714	0.643	0.612	0.619	0.763	0.779	0.768
Linear SVM (C=1)	0.715	0.641	0.635	0.631	0.769	0.766	0.764
RBF SVM	0.619	0.708	0.049	0.091	0.617	0.984	0.759
Neural Net	0.686	0.656	0.485	0.528	0.723	0.815	0.757
Gradient Boosting Classifier	**0.736**	0.673	0.641	**0.653**	0.777	0.796	**0.785**

Table 1: Classification results for the 1^{st} experiment with different algorithms and parameter settings

For detailed results of the 2^{nd} experiment see Table 2.

[11] Support Vector Classifier class in *sklearn*
`http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

Classifier	**Accuracy**	**Precision (+ class)**	**Recall (+ class)**	**F-score (+ class)**	**Precision (- class)**	**Recall (- class)**	**F-score (- class)**
Baseline	0.835	0.000	0.000	0.000	0.835	1.000	0.910
Linear SVM (C=0.025)	0.984	0.964	0.937	0.950	0.988	0.993	0.990
Linear SVM (C=1)	0.989	0.968	0.966	0.967	0.993	0.994	0.993
RBF SVM	0.947	1.000	0.679	0.805	0.940	1.000	0.969
Neural Net	0.982	0.939	0.953	0.946	0.991	0.987	0.989
Gradient Boosting Classifier	**0.993**	0.984	0.973	**0.978**	0.995	0.997	**0.996**

Table 2: Classification results for the 2^{nd} experiment with different algorithms and parameter settings

7. Conclusion and Future Work

The method described in this paper is aimed at solving supervised classification task on short Serbian messages. In order to solve this task in supervised manner, it is important to have representative corpus of SMS data and metadata such as sender's name, phone number, etc. Due to privacy concerns, people are having trust issues and are not willing to share their SMS messages. As a consequence, evaluation of the method developed in this paper is not performed on other datasets with the same structure and content. Twitter data might seem as a good candidate (Twitter corpora are publicly available, there is the same character count threshold and there is plenty of it), but Twitter posts and SMS messages are not having the same purpose. SMS message is addressed for specific person and most often asks question or answers one. Twitter posts mostly contain opinions or comments, referring to other users or topics using hash tags. These hash tags are very common in tweets and can be a rich source of even more text features. Although the problem itself could be stated on any type of text that is interchanged between two or more sides (Facebook posts, tweets, E-mails, SMS messages, forum posts, Viber/WhatsApp messages etc.), it is expected that, due to difference in purpose of these different services, different approach should be applied for each.

Examining only emoticons, punctuation usage or abbreviations is not enough to identify a person. Even for a human, it would be impossible to tell difference between persons who are writing with perfect grammar and without emoticons. But with additional information like one used in (Laperdrix et al., 2016) and (Eckersley, 2010), this task might be simple. In the future work, it is intended to generalize the problem so Facebook and Twitter posts can be evaluated. This is primarily aimed at enriching model with new features, such as message semantics (word meanings, context, used language dialect and chat history), sender's gender, common phrases used by a sender and even information about the device from which the message is sent (e.g. the device model or underlying operating system).

At the time being, current results are implying that this kind of identification is possible, at least as one of the steps in the authorship attribution.

Acknowledgements

This research was supported by the Serbian Ministry of Education and Science under grant #178006.

References

Abbasi, A. and Chen, H. (2008). Writeprints: A Stylometric Approach to Identity-level Identification and Similarity Detection in Cyberspace. *ACM TOIS*, 26(2):1–29.

Abbasi, A., Chen, H., and Nunamaker, J. (2008). Stylometric Identification in Electronic Markets: Scalability and Robustness. *Journal of Management Information Systems (JMIS)*, 25(1):49–78.

Derks, D., Bos, A. E., and Von Grumbkow, J. (2007). Emoticons and Social Interaction on the Internet: the Importance of Social Context. *Computers in human behavior*, 23(1):842–849.

Ebert, S. (2017). *Artificial Neural Network Methods Applied to Sentiment Analysis*. Ph.D. thesis, Ludwig-Maximilians-Universität München.

Eckersley, P. (2010). How Unique is your Web Browser? In *Proceedings of the* 10^{th} *International Conference on Privacy Enhancing Technologies*, volume 6205, pages 1–18. Berlin, Heidelberg: Springer - Verlag.

Friedman, J. H. (2001). Greedy Function Approximation: a Gradient Boosting Machine. *Annals of statistics*, pages 1189–1232.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). Springer, 2 edition.

Laperdrix, P., Rudametkin, W., and Baudry, B. (2016). Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints. In 37^{th} *IEEE Symposium on Security and Privacy (S&P 2016), San Jose, United States*, pages 878–894. IEEE.

Mairesse, F., Walker, M., Mehl, M., and Moore, R. (2007). Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *Journal of Artificial Intelligence Research (JAIR)*, 30:457–500.

Orebaugh, A. and Allnutt, J. (2009). Classification of Instant Messaging Communications for Forensics Analysis. *Social Networks*, pages 22–28.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pennebaker, J. and King, L. (1999). Linguistic Styles: Language Use as an Individual Difference. *Journal of Personality and Social Psychology*, 77(6):1296–1312.

Read, J. (2005). Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification. In *Proceedings of the ACL student research workshop*, pages 43–48. Association for Computational Linguistics.

Repar, A. and Pollak, S. (2017). Good Examples for Terminology Databases in Translation Industry. In *eLex 2017: eLex 2017: The* 5^{th} *biennial conference on electronic lexicography, Netherlands, 19-21 September 2017*, pages 651–661.

Roffo, G., Giorgetta, C., Ferrario, R., and Cristani, M. (2014). Just the Way You Chat: Linking Personality, Style and Recognizability in Chats. In *International Workshop on Human Behavior Understanding*, pages 30–41. Springer.

Stamatatos, E. (2009). A Survey of Modern Authorship Attribution Methods. *Journal of the Association for Information Science and Technology (JASIST)*, 30(3):538–556.

Škorić, M. (2017). Classification of Terms on a Positive-negative Feelings Polarity Scale Based on Emoticons. *Infotheca: Journal for Digital Humanities*, 17(1):67–91.

Walther, J. B. and D'Addario, K. P. (2001). The Impacts of Emoticons on Message Interpretation in Computer-Mediated Communication. *Social science computer review*, 19(3):324–347.

Zheng, R., Li, J., Chen, H., and Huang, Z. (2006). A Framework for Authorship Identification of Online Messages: Writing-style Features and Classification Techniques. *Journal of the Association for Information Science and Technology (JASIST)*, 57(3):378–393.