Gender Prediction for Chinese Social Media Data

Wen Li
Department of Linguistics
Indiana University
Bloomington, IN, USA
wl9@indiana.edu

Markus Dickinson
Department of Linguistics
Indiana University
Bloomington, IN, USA
md7@indiana.edu

Abstract

Social media provides users a platform to publish messages and socialize with others, and microblogs have gained more users than ever in recent years. With such usage, user profiling is a popular task in computational linguistics and text mining. Different approaches have been used to predict users' gender, age, and other information, but most of this work has been done on English and other Western languages. The goal of this project is to predict the gender of users based on their posts on Weibo, a Chinese micro-blogging platform. Given issues in Chinese word segmentation, we explore character and word *n*-grams as features for this task, as well as using character and word embeddings for classification. Given how the data is extracted, we approach the task on a per-post basis, and we show the difficulties of the task for both humans and computers. Nonetheless, we present encouraging results and point to future improvements.

1 Introduction and Motivation

Author profiling, the task of determining some demographic property of a language user (gender, age, personality, etc.), has become a significant area within NLP and text mining, with many practical applications (see, e.g., Rangel et al., 2015), and it links to a bevy of related subfields (authorship attribution, native language identification, sentiment analysis, etc.) (cf. Argamon et al., 2009), in that they share in common methods and features (e.g., lexica, *n*-grams). Despite obtaining promising results in certain tasks with certain data sets (e.g., Schler et al., 2006), the challenges in author profiling increase as the text gets shorter and noisier, as with social media data, given that there seem to be fewer and less reliable indicators of a demographic trait (e.g., Zhang and Zhang, 2010; Burger et al., 2011), in addition to the fact that many users produce language atypical of their demographic (Bamman et al., 2014; Nguyen et al., 2014). This problem is potentially compounded when examining languages such as Chinese, where: a) the definition of a word is problematic (Sproat et al., 1996); b) the collection of data with links to individual users is challenging, since Weibo (see below) requires users' authorization before data collection; and c) there has been no published work (we are aware of) on this task, most work focusing on English and to some extent other Western languages (Rangel et al., 2015; Nguyen et al., 2013).

We set as our first step that of predicting the gender of users on Weibo[1]—the Twitter analogue in China—based on an individual post. This task reveals two sub-goals. First, we want to identify areas of development in moving to Chinese social media data. Having no work on identifying gender or other author characteristics in Chinese is a pity, as previous work has debated the importance of character-based *n*-grams vs. word-based *n*-grams (cf., e.g., Sapkota et al., 2015), and Chinese, with greater difficulty in word segmentation, is an excellent proving ground for such issues. Indeed, although there are many Chinese syntactic issues to deal with (e.g., the nominal classification system), we focus our attention on the impact of different kinds of *n*-gram models. This is particularly relevant in Chinese as characters in the logographic (meaning-based) Chinese writing system mean something different than characters in alphabetic systems, and with a larger number of characters there will be sparser *n*-grams. As a side note,

[1] http://weibo.com

Proceedings of Recent Advances in Natural Language Processing, pages 438–445,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_058

there is work utilizing Weibo for word segmentation (e.g., Zhang et al., 2013) and sentiment analysis (e.g., Zhou, 2015); with our work we pave the way for future connections by exploring the impact of data filtering, preprocessing, and *n*-gram features on system performance.

A second sub-goal is to identify specific difficulties and specific opportunities with a per-post (vs. per-user) method of classification, as this means we have very little data with which to work, as little as a few words (see section 2). Burger et al. (2011) note that their "tweet text classifier's accuracy increases as the number of tweets from the user increases," and Nguyen et al. (2014) point out cases where features associated with one gender are found in tweets of the opposite gender. We assess how accurate such a classifier can be, for humans or machines, and the impact of the choice of features on classification accuracy. Indeed, we find automatic classification accuracy no higher than 63% on this per-post task (section 4.1)—also true for human accuracy (section 4.3)—and our work suggests that per-post classification should focus on the identification of posts which can be reliably classified rather than on improving overall accuracy (sections 4.2 and 4.4).

Given short messages and associated data sparsity, we take the additional step of investigating the role that semantic similarity methods can have in classification. The popularity of `word2vec` in recent years comes from the weakness of traditional bag-of-words model: words are represented as isolated indices, and the vectors to represent a document are often sparse (Mikolov et al., 2013). As mentioned in Mikolov et al. (2013), `word2vec` captures some syntactic regularities and semantic similarities. Some research is starting to use `word2vec` for text classification of Chinese texts, especially for sentiment analysis (Su et al., 2014; Bai et al., 2014; Zhang et al., 2015), and we would like to explore the use of `word2vec` techniques in the gender classification task. Because of the attributes of Chinese characters, the "word" vectors could be built based on characters or words. While most approaches train the vectors based on Chinese words, with word segmentation done beforehand, there is also research using character representations (Sun et al., 2014). However, the difference between character and word vectors in Chinese is not clear, and no one has conducted a detailed comparison between them for text classification. We would thus like to help fill this gap.

Users are required to specify their gender on Weibo, giving good experimental data (section 2), but making the task less immediately useful.[2] The task is still worth pursuing because the insights are applicable for predicting other demographics (e.g., age) and for (current or future) data beyond Weibo.

2 Data

Weibo (also known as Sina Weibo) is a Chinese microblogging site, with a market penetration similar to the United States' Twitter. According to Wikipedia,[3] as of the third quarter of 2015, Weibo has 222 million subscribers and 100 million daily users. About 100 million messages are posted each day on Weibo.

Weibo implements many features from Twitter. A user may post with a 140-character limit, mention or talk to other people using `@UserName` formatting, add hashtags with `#HashName#` formatting, follow other users to make their posts appear in one's own timeline, re-post with `//@UserName` similar to Twitter's retweet function `RT @UserName`, and select posts for one's favorites list. The users of Weibo include Asian celebrities, movie stars, singers, famous business and media figures, as well as some famous foreign individuals and organizations; like Twitter, Sina Weibo has a verification program for known people and organizations.

URLs are automatically shortened using the domain name `t.cn` like Twitter's `t.co`. Official and third-party applications make users able to access Sina Weibo from other websites or platforms. In January 2016, Sina Weibo decided to remove the 140-character limit for any original posts, and users were thereby allowed to post with up to 2000 characters, while the 140-character limit was still applicable to re-posts and comments.

2.1 Collection

We collected random Weibo users' posts in February and March 2015, using the Weibo developer API.[4] The API allows one to get 200 recent public posts without user authorization. Since short lag times might lead to duplicate documents, we

[2] Users may of course falsely report their gender, leaving some noise in the data; an accurate classifier may in the long run help pinpoint such misreporting.

[3] `https://en.wikipedia.org/wiki/Sina_Weibo`

[4] `http://open.weibo.com`

called the API every three minutes. Thus, the collected data are organized by time (i.e., per-post), not by user.

2.2 Filtering

We filter posts from users with more than 500,000 followers, since these accounts are often maintained by organizations or public relations teams (e.g., for celebrities), which produce very different contents than for ordinary users. We also try to filter posts containing headline news, usually starting with "【".

Some posts on Weibo are written in languages other than Chinese, so we only keep posts with at least 70% Chinese characters in them (punctuation counting as valid Chinese here).

There are some additional challenging posts to filter, namely those automatically generated by third-party applications or ones trying to sell products. Such formulaic posts often result in highly similar contents to each other, overweighting terms, and the notion of user demographics is unclear for them. After some initial examination, we remove these posts using keywords and keyphrases such as "我参加了 (I participated in)" and "请点击 (please click on)". It should be noted, however, that some of the words and phrases used for filtering may occur in posts actually written by users, so we run some risk of overfiltering.

2.3 Summary

After filtering, we use 50,000 posts (2.2 million characters) for classification: 45,000 as training data, 5,000 as test data. 28,901 (57.8%) of the posts are written by female users, 21,099 (42.2%) by males. The average post length is 42 characters. We use 100,000 random posts to train character and word vectors.

3 Methods

3.1 Preprocessing

For preprocessing, we first normalize the data for obtaining more reliable *n*-grams, by: a) removing all the URLs; and b) replacing Weibo-exclusive emoticons and emojis with a single character that exists nowhere else in the data. In initial experiment, we tried to keep the top 50 most frequent emoticons and emojis, replacing all the others, which resulted in a 0.3% decrease in accuracy.

Secondly, to test different kinds of features, we segment the data into words, using the Stanford Chinese Word Segmenter (Tseng et al., 2005). Although developed for well-edited data, hand-examination reveals no major issues for posts; importantly, the segmenter is at least consistent across posts.

	Char.	Word
Uni.	6,463	81,280
Bi.	333,748	517,639
Tri.	987,781	877,571
Total	1,327,992	1,476,490

Table 1: Number of *n*-grams in training

3.2 Features

3.2.1 *n*-gram features

We first focus on features that allow us to explore the impact of segmentation. Starting with **character-based *n*-grams**, we first extract all unigrams, bigrams, and trigrams in the training data; numbers are in Table 1 for the cleaned data.

We then use information gain (Liu et al., 2014) to select the top 10,000 *n*-grams, from the set of: a) unigrams (*Uni.*),[5] b) bigrams (*Bi.*), c) trigrams (*Tri.*), d) unigrams and bigrams (*Uni.*+*Bi.*), or e) all three (*All*). For condition d (for the cleaned data), among the 10,000 features, there are 1,122 unigrams and 8,878 bigrams; for condition e, there are 842 unigrams, 4,698 bigrams, and 4,460 trigrams.

We then do the same for the **word-based *n*-grams**; for condition d, we obtain 3,298 unigrams and 6,702 bigrams; for condition e, there are 2,526 unigrams, 4,501 bigrams, and 2,973 trigrams. The features are binary, reflecting *n*-gram presence/absence.

We currently do not mix character and word-based *n*-grams, to mitigate the effect of feature overlap in determining utility. Taking an example from English, the character trigram *the* overlaps with word unigrams *the* and *them*, among others. Future work could explore mixing.

3.2.2 Word embeddings

We additionally focus on generalizing beyond simple characters and words by using word vectors; as mentioned in section 2.3, we train the word/character vectors on 100,000 posts. While

[5] Since there are fewer than 10,000 character unigrams, we use all 6,643 of them.

training character vectors does not require any preprocessing, word vectors require Chinese word segmentation. To obtain a feature vector for an entire post, we add up the word/character vectors according to the words/characters occurring in a Weibo post and divide the sum by the number of words/characters occurring in the post.

We use Gensim[6] to train our vectors. There are a number of parameters available, and for some of the parameters, there is no intuitive clue of what values would better work for this task. Since we could not try all the combinations exhaustively, in this project we train both continuous bag-of-words (*CBOW*) and skip-gram (*SG*) models; keep the context window size as 5; set the minimum frequency of a word/character as 3; and vary dimensions among *100d*, *200d*, and *500d*. All other parameters are as default settings. For word vectors, there are 40,561 distinct words in the vocabulary; for character vectors, there are 6,581 character types in the vocabulary.

3.3 Classifiers

We use different classifiers implemented by `scikit-learn` (Pedregosa et al., 2011). For classification with *n*-gram features, we initially employed Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), and Logistic Regression, and found differences that were less than 0.5% in accuracy, with MNB performing the best. We then decide to use MNB for the *n*-gram experiments, setting α to 0.01.

Since MNB does not work with `word2vec` features—as some values are negative in the feature vector—we use Random Forest for classification with character/word vectors.

4 Evaluation

4.1 Results

4.1.1 Character-based *n*-grams

Noisy data We first run a MNB classifier with default settings on partly filtered data, namely before using keyword filtering (section 2.2), using character-based unigrams, bigrams, and trigrams. This achieves an overall accuracy of 63.2%; Table 2 shows precision and recall values for the genders. Note that the classifier guesses female 3,085 times and male 1,915.

The results should be taken with a grain of salt, as much of the noisy data has repeated patterns in it; for example, females are more likely to have "我参加了 (I participated in)", but these come from auto-generated messages after participating in, e.g., contests and events that have a change to win some prizes, and males are more likely to have "请点击 (please click on)", but these seem largely to be advertisements.

	Acc	Prec	Rec
Female	n/a	66.7	71.6
Male	n/a	57.4	51.7
Avg.	63.2	62.8	63.2

Table 2: Results on noisy data (%), with (10,000) character-based *n*-grams (*All* model): Test: 2,875 F, 2,125 M.

Cleaned data Results of character *n*-gram models on cleaned data are given in the left side of Table 3. The best model uses *All* *n*-gram types, with an overall accuracy of 62.8% and a classifier distribution of 3,289 females and 1,711 males. Individually, unigrams perform better than bigrams or trigrams. Comparing *All* to *Uni.*, some improvement comes from the guessing of males.

4.1.2 Word-based *n*-grams

The better guessing of females, likely due to more salient features for females (see section 4.4), is repeated with word-based *n*-grams, as in the right side of Table 3. Overall accuracy of the *All* model is about the same as with character-based *n*-grams, 62.8%, but with more bias towards females: 3,400 female guesses vs. 1,600 male.

4.1.3 Word embeddings

The accuracy of gender classification using `word2vec` with different settings is shown in the bottom part of Table 3. We can see that in general, word vectors perform better than character vectors, and the *CBOW* model works better than the *SG* model for Chinese data. For the *CBOW* model, increasing the dimension helps accuracy, while the *SG* model performs better with lower dimension vectors.

We plan to explore better `word2vec` training data and different ways of incorporating this information, in addition to a wider range of features. It seems, however, that all of our models are hitting a ceiling, accuracy-wise, leading us to pursue a different approach to the problem.

[6] https://radimrehurek.com/gensim/index.html

	Char.					Word				
		Female		Male			Female		Male	
Model	Acc	Prec	Rec	Prec	Rec	Acc	Prec	Rec	Prec	Rec
Uni.	62.0	65.2	74.6	55.6	44.3	61.6	65.3	72.9	54.8	45.9
Bi.	61.7	65.1	74.0	55.0	44.5	61.4	63.8	78.2	55.5	38.0
Tri.	60.7	62.6	81.2	55.0	32.1	60.6	61.5	86.6	56.3	24.3
Uni.+Bi.	62.1	65.9	72.5	55.3	47.5	62.6	65.9	74.1	56.2	46.5
All	**62.8**	66.1	74.5	56.6	46.5	**62.8**	65.6	76.4	57.0	43.7
CBOW_100d	60.9	63.1	78.2	55.2	37.0	62.2	64.1	79.2	57.3	38.7
CBOW_200d	61.0	63.2	78.0	55.3	37.5	62.0	64.1	78.7	56.8	38.9
CBOW_500d	61.2	63.0	80.8	56.1	35.3	**62.9**	64.6	79.9	58.6	39.5
SG_100d	60.8	63.2	77.2	54.9	38.2	62.6	64.4	79.8	58.1	38.8
SG_200d	61.2	63.3	78.7	55.9	37.1	62.5	64.3	79.5	57.8	38.8
SG_500d	60.9	62.8	80.1	55.8	34.6	61.7	63.8	79.1	56.5	37.7

Table 3: Results on cleaned data (%), with *n*-gram features and `word2vec` features. Training: 45k; Test: 5k.

4.2 More Accurate Cases

Confidence of prediction With very little text, we can use additional information to identify cases for which the classifier is more accurate. Because the classifier assigns a score between 0 (female) and 1 (male), our definition of **confidence** corresponds to the distance from the midpoint (0.5), which ranges from 0.0 to 0.5. A confidence of 0.29, for example, means that the classifier assigned a score of either 0.21 or 0.79.

Post length Within the limit of 140 (Chinese) characters, the Weibo post lengths vary greatly. While a shorter post may not contain enough information for the classifier to make a correct prediction, a longer post is more likely to be a paragraph of famous quotes or a short story, confusing the classifier due to the lack of gender indicators. With regard to character-based and word-based settings, we define the **length of a post** as the number of characters or words, respectively.

Quartiles (Q_1, Q_2, Q_3) of confidence and length for character-based and word-based *All* model are shown in Table 4, with accuracies for the corresponding intervals in Table 5. For example, for character-based confidence, we report an accuracy of 61.2% for confidences between Q_1 (0.29) and Q_2 (0.47), i.e., either in the range (0.03, 0.21) or (0.79, 0.97). By narrowing in on high-confidence cases or posts with sufficient information (i.e., 16–40 words), we can obtain accuracy around 70%. Running the same experiments with `word2vec` features displayed the same trends.

	Confidence		Length	
	Char.	Word	Char.	Word
Q_1	0.2893	0.2357	13.0	9.0
Q_2	0.4726	0.4352	25.0	16.0
Q_3	0.4998	0.4986	62.0	40.0

Table 4: Quartiles for confidence/length of *All* model.

	Confidence		Length	
Interval	Char.	Word	Char.	Word
$\leq Q_1$	57.2	54.3	57.0	60.4
$(Q_1, Q_2]$	61.2	60.5	68.4	63.2
$(Q_2, Q_3]$	63.2	67.8	**68.7**	**70.1**
$> Q_3$	**69.2**	**68.6**	57.3	57.4

Table 5: Accuracy (%) for confidence/length quartiles of *All* model.

4.3 Human Judgment

We asked four people to independently guess the gender of 200 random Weibo posts. The accuracies of two Weibo users are 64.0% (*user_M*) and 59.5% (*user_F*), while the two people who do not use Weibo obtain accuracies of 58.5% (*non_user_F*) and 55.5% (*non_user_M*). For comparison, the character-based, all *n*-gram classifier achieves an accuracy of 64.5% on the same 200 posts. The detailed results are shown in Table 6.

Not only are the humans no better than automatic classification, but we observe the same tendency of predicting more females than males, with the humans also better at recognizing females than

		Female		Male	
	Acc	Prec	Rec	Prec	Rec
user_M	64.0	65.7	78.3	60.3	44.7
user_F	59.5	63.3	70.4	52.8	44.7
non_user_M	55.0	61.5	58.3	47.3	50.6
non_user_F	58.5	64.0	63.5	51.2	51.8

Table 6: Human judgment results (%) for Weibo (*M*/*F* = male/female annotator)

males.

4.4 Discussion

We have seen an overall per-post accuracy of approximately 62.8%. Interestingly, this is true regardless of whether it is a model of character-based or word-based *n*-grams, despite relying on an automatic segmenter. The unigram models perform better individually than either the bigram or trigram models, likely due to fewer issues with sparsity—particularly important for individual posts. Relying on confidence or length can boost accuracy, up to nearly 70%. It is important to note that higher reported performances in previous work deal with per-user classification tasks; more comparably, Burger et al. (2011) report an accuracy of around 64% for per-tweet (per-post) gender classification. Some researchers mentioned using ensemble classifiers to improve the accuracy for text classification tasks (Liu et al., 2016; Li and Zou, 2017), which could be future work for this gender prediction task.

Given that humans also perform with around 60% accuracy, one tentative conclusion is that users only post like their gender about 60% of the time. If true, this may be key for moving from per-post classification to a per-user aggregation.

Additionally, these results support the intuition that it is going to be virtually impossible to correctly classify every post. The quest for identifying posts which can be more accurately classified, as in section 4.2, becomes more important: instead of trying to boost overall accuracy—which various feature settings have failed to do—the important per-post question may be, can one reliably classify some significant portion of the data and identify such a portion automatically? This idea of not attempting to classify every post may thus be useful for a per-user classification, as unreliable posts could distract from an overall trend.

Most important features Classification is better for females than males: as with some previous work (e.g., Burger et al., 2011), this seems attributable to more salient features in female posts. Our MNB classifier does not provide the most important features used in classification, but we use information gain to examine the top 300 features (both character-based and word-based) by hand.

Among the top 40 features, most of them are punctuation marks or repeated characters/emoticons indicating femaleness, such as "！！", "～", "啊啊啊 (ah ah ah)", "嘤嘤嘤 (*sound of sobbing*)", "哈哈哈 (ha ha ha)". Dozens of content words/phrases are in the top 300, such as "开学 (school starts)", "头发 (hair)", "我妈 (my mom)", "姐姐 (sister)", "不开心 (unhappy)", etc. We also observe the names of "鹿晗 (Lu Han)", "王俊凯 (Wang Junkai)", and "TFBOYS" in the list, which indicates the comprehensive popularity of these younger stars on Weibo, especially among female users. Named entity classification may be of future help.

One issue in determining important features from a small set of posts is how to handle multiple instances of the same *n*-gram within the same file. One may wish to count instances or, alternatively, to normalize repetitive characters or words into single instances. Initial results of incorporating *n*-gram counts show a slight drop in performance (cf., e.g., Burger et al., 2011).

5 Summary and Outlook

We have set about predicting the gender of users based on their posts on the Chinese micro-blogging platform Weibo. Given issues in Chinese word segmentation, we have explored character and word *n*-grams as features for this task, as well as using character and word embeddings for classification, and we have shown that all models perform near the same level. With humans performing with the same accuracy on a per-post basis, we have seen a need to explore the identification of high-confidence classification cases. In short, adapting techniques from English to Chinese for identifying gender on social media does not generally seem problematic, as character and word *n*-gram models seem equally effective. Classifying on a per-post basis, however, requires much more investigation, as well as using the information to move to per-user models.

For short posts, we need to explore ways of link-

ing similar content across lexical variation, such as by incorporating distributional semantic representations (e.g., Ji and Eisenstein, 2013) or paraphrase identification (e.g., Preoţiuc-Pietro et al., 2016). Using aggregate features—post length, presence of sentence-ending particles, emoticon categories, etc.—should also help in this, as well as more data cleaning (e.g., removal of duplicate posts).

References

S. Argamon, M. Koppel, J. Pennebaker, and J. Schler. 2009. Automatically profiling the author of an anonymous text. *Communications of the ACM* 52(2):119–123.

X. Bai, F. Chen, and S. Zhan. 2014. A study on sentiment computing and classification of sina weibo with word2vec. In *2014 IEEE International Congress on Big Data (pp. 358-363). IEEE*.

David Bamman, Jacob Eisenstein, and Tyler Schnoebelen. 2014. Gender identity and lexical variation in social media. *Journal of Sociolinguistics* 18(2):135–160.

John D. Burger, John Henderson, George Kim, and Guido Zarrella. 2011. Discriminating gender on twitter. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK., pages 1301–1309.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*. Seattle, WA, pages 891–896.

Wen Li and Liang Zou. 2017. Classifier Stacking for Native Language Identification. In *Proceedings of the 12th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Copenhagen, Denmark.

Can Liu, Sandra Kübler, and Ning Yu. 2014. Feature feature selection for highly skewed sentiment analysis tasks. In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP), pages 2–11, Dublin, Ireland*.

Can Liu, Wen Li, Bradford Demarest, Yue Chen, Sara Couture, Daniel Dakota, Nikita Haduong, Noah Kaufman, Andrew Lamont, Manan Pancholi, Kenneth Steimel, and Sandra Kübler. 2016. IUCL at SemEval-2016 Task 6: An Ensemble Model for Stance Detection in Twitter. In *Proceedings of SemEval-2016*. San Diego, California, pages 394–400.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781*.

Dong Nguyen, Rilana Gravel, Dolf Trieschnigg, and Theo Meder. 2013. "how old do you think i am?"; a study of language and age in twitter. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*. AAAI Press, Palo Alto, CA.

Dong Nguyen, Dolf Trieschnigg, A. Seza Doğruöz, Rilana Gravel, Mariet Theune, Theo Meder, and Franciska De Jong. 2014. Why gender and age prediction from tweets is hard: Lessons from a crowdsourcing experiment. In *Proceedings of COLING*

2014, the 25th International Conference on Computational Linguistics: Technical Papers. Dublin, Ireland, pages 1950–1961.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Daniel Preoţiuc-Pietro, Wei Xu, and Lyle Ungar. 2016. Discovering user attribute stylistic differences via paraphrasing. In *Proceedings of AAAI 2016*.

Francisco Rangel, Fabio Celli, Paolo Rosso, Martin Pottast, Benno Stein, and Walter Daelemans. 2015. Overview of the 3rd author profiling task at pan 2015. In Linda Cappelato, Nicola Ferro, Gareth Jones, and Eric San Juan, editors, *CLEF 2015 Labs and Workshops, Notebook Papers*. Toulouse, France, CEUR Workshop Proceedings.

Upendra Sapkota, Steven Bethard, Manuel Montes, and Thamar Solorio. 2015. Not all character n-grams are created equal: A study in authorship attribution. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, CO, pages 93–102.

J. Schler, Moshe Koppel, S. Argamon, and J. Pennebaker. 2006. Effects of age and gender on blogging. In *Proceedings of the AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*.

Richard Sproat, William Gale, Chilin Shih, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for chinese. In *Computational linguistics 22, no. 3 (1996): 377-404*.

Z. Su, H. Xu, D. Zhang, and Y. Xu. 2014. Chinese sentiment classification using a neural network tool—word2vec. In *Multisensor Fusion and Information Integration for Intelligent Systems (MFI), 2014 International Conference on (pp. 1-6). IEEE*.

Y. Sun, L. Lin, N. Yang, Z. Ji, and X. Wang. 2014. Radical-enhanced chinese character embedding. In *International Conference on Neural Information Processing (pp. 279-286). Springer International Publishing*.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*.

Cathy Zhang and Pengyu Zhang. 2010. Predicting gender from blog posts. Technical report, University of Massachusetts, Amherst.

D. Zhang, H. Xu, Z. Su, and Y. Xu. 2015. Chinese comments sentiment classification based on word2vec and svm perf. *Expert Systems with Applications* 42(4):1857–1863.

Longkai Zhang, Li Li, Zhengyan He, Houfeng Wang, and Ni Sun. 2013. Improving chinese word segmentation on micro-blog using rich punctuations. In *ACL (2), pp. 177-182*.

Hongzhao Zhou. 2015. Rule-based weibo messages sentiment polarity classification towards given topics. In *ACL-IJCNLP 2015 (2015): 149*.