university of groningen / faculty of arts

Detecting Income Level of Dutch Twitter Users using Stylometric Features

Léon Melein

Bachelor thesis - Concept version
Information Science
Léon Melein
S2580861
December 20, 2016

ABSTRACT

To be written on Tuesday.

CONTENTS

PREFACE

To be written on Tuesday.

1 INTRODUCTION

As we rely more and more on the internet and its applications in our daily lives, the opportunities for author profiling continue to grow. Social media services, and the text-based communications they facilitate, provide an ever-growing corpus of texts, which are linked to their author. Furthermore, these services provide additional self-disclosed information about the authors like education, occupation and relationships. If we use this data the right way, we can perform research that simply wasn't possible before or on a much bigger scale than was possible before (Sloan et al., 2015).

An aspect of author profiling is income prediction. A potential use for these estimations of income is in customer relations. Just as the rest of our life, our contact with companies and their customer services moves online. People want to be helped quickly and without giving a lot of information in advance. They expect companies to "know" them. Companies want to know their customers so they don't have to ask for a lot of information and can help quickly. Income can be a key factor in this, because it can help in determining which products or services a customer probably already has and in which it is likely interested.

An example: a cable company customer with a relatively low income probably has a low tier service package. He or she will only be interested in cheap(er) options or options that provide significant extra value, in comparison to their additional cost. Knowing this, offering an expensive movie channel package to this customer does not make a lot of sense. Such a package is nice to have, but isn't of much value - unless the customer is a true movie addict.

Despite the existing research into English speaking Twitter users and their respective incomes, there is currently no comparable research for Dutch speaking Twitter users. We will therefore focus our research on the following question: to what extent is it possible to accurately predict the income level of a Dutch Twitter user and which stylometric features are predictive?

In the following chapter, we will look at the existing body of research for English Twitter users. Afterwards, we will detail our data gathering efforts, after which our research method is detailed. We then discuss the results gathered and conclude tot what extent we can accurately predict income level.

2 BACKGROUND

Research into author profiling on Twitter is relatively new, especially in the field of income prediction. All prior work has focused on English-speaking Twitter users.

The most recent study was performed by Flekova et al. (2016). Their goal was to find a viable writing style-based predictor for age and income. For each dimension, a different data set was used. Regarding income, an off the self data set was used, which will be discussed in more detail later on. It contained almost six thousand Twitter users labeled with their occupation. The researchers used the occupations to label the users with the mean UK yearly income for their occupation. This was done regardless of the location of the users involved (e.g., an English-speaking user from Amsterdam would also be assigned a UK income). As the exact income for every occupation was known, the machine learning task was framed as regression. Flekova et al. codified stylistic variation into a number of features, which were grouped into four categories: surface, readability, syntax and style. After performing a ten-fold cross validation with both linear and non-linear regression methods they discovered that readability metrics like the *Flesch Reading Ease* metric and the relative use of pronouns correlated with income over age ($r_{income} = .297, r_{Flesch} = .315; p < 0.001$). They concluded that the differences in style can be used to "tailor the style of a document without altering the topic to suite either age or income individually".

The data set used in Flekova et al. (2016) was created during an earlier study by Preotiuc-Pietro et al. (2015). They used the corpus to classify users according to their occupational class. The occupational titles and classes used were gathered from the UK Standard Occupational Classification (SOC) (Office for National Statistics, 2010). The SOC is a hierarchical classification of occupations. It has four levels, starting with nine very general classes and terminating in hundreds of very specific classes. Each level is indicated with a different number of digits. The coarsest level is indicated with one digit and the finest level with four digitals (e.g., class 1: 'managers, directors and senior officials' and class 1116: 'elected officials and representatives', respectively). The classification is based on the International Standard Classification of Occupations (International Labor Office, 2013). For each occupation they used the Twitter REST API to find at most 200 users for each occupation. The accumulated users were divided into the three-digit groups they belong to. Users that were companies, had no description or had a contradicting description were removed from the collection by hand. Furthermore, three-digit groups with less than 45 users were discarded. The final collection contained 5191 users divided into 55 three-digit groups.

Preotiuc-Pietro et al. mention two papers in their related work section that describe different labeling strategies. Both do not annotate users manually. They all employ *distant supervision*, a method that labels data automatically, given an existing form of ground truth. In the Preotiuc-Pietro study, Twitter users were labeled by looking for a self-disclosed occupational title. In order to detect such titles they relied on a list of occupational titles from the SOC. This list forms the ground truth needed for labeling.

The first is a paper by Li et al. (2014) which tries to label users with the name of their employer, among other things. In this study, Twitter profiles were cross-linked with profiles on a different social networking site, Google Plus. To ensure the both profiles are interrelated they looked at the friend connections on both sites and made sure that there was a large enough intersection between them. The employer name was extracted from the Google Plus profile and used to label the Twitter user. Using this method they were able to collect 7208 users with a known employer. This strategy relies heavily on profiles from networking sites other than Twitter, like Google Plus or LinkedIn. Without unfettered access or a large number of cross-linkable profiles on both platforms, it cannot be used for labeling data.

The second strategy only relies solely on the occupational titles from the SOC and profiles on Twitter. Sloan et al. (2015) labeled users with class in the National Statistics - Socio-Economic Classification (NS-SEC), a classification closely related and interoperable with the SOC. They extracted users from a feed provided by the Twitter API, which constitutes a 1 percent representative sample of public tweets. For each user, they looked for an self-disclosed occupational title in the biography line. The titles they used for detection were gathered from the SOC. After finding an occupation, the user would be labeled with their NS-SEC class by looking up the SOC class for their occupation and then looking up the corresponding NS-SEC class. In case a user mentioned multiple occupations, the authors hypothesized that the first one found would be most important and therefore should be used. Using this method they were able to label 32032 users with their NS-SEC class. A random survey of 1000 users resulted in an accuracy of 57,8 percent. The authors mention several caveats of this method. Hobbies and former occupations may be falsely detected as current occupations. Commonly occurring phrases like "Doctor Who fan" may also result in false matches.

This second strategy depends a lot less on outside data. It is therefore easier to implement and use in further research, given that the needed data is available. For this thesis, a corpus of Dutch tweets and their authors available is already at the University of Groningen and data on occupational titles and incomes is available from the Dutch government bureau Statistics Netherlands. It therefore makes sense to use this strategy for our data collection and annotation efforts. The implementation details of this strategy follow in the next chapter.

3 DATA AND MATERIAL

3.1 COLLECTION

The primary data set for this research is a corpus of Dutch Twitter users with their 500 latest tweets, categorized on income class. As there was no suitable data set available off the shelf, a new corpus was created. As a starting point, the University of Groningen (UG) twitter2 corpus was used to gather user profiles. The twitter2 corpus contains all Dutch tweets provided by Twitter's Streaming API, which constitutes a 1 percent representative sample of public messages posted on Twitter.

In order to gather user profiles, we used all tweets from september 1th till september 5th, 2016. For each tweet in the corpus, we looked up its user by using the UG's in-house *tweet2tab* tool to extract the user ID, username, real name and biography line for each user from the corpus. The user's biography line was used to find an occupational title. That title was then linked to an occupational class and consequently the average hourly income for that occupational class. The average hourly income was then multiplied by the average number of worked hours in the Netherlands to compute the average yearly income. All users with a known occupation were labeled with their average yearly income. This resulted in a collection of 36113 users with known occupations and incomes.

After removing no longer existing accounts, private accounts and accounts with less than 1000 tweets, 21862 users were still available. These users were divided into two income classes, high (above €34.500) and low (below €34.500). Afterwards, 1500 users were randomly selected from each group and their tweets were gathered using the Twitter API. Retweets and non-Dutch tweets (as explained in the next section) were left out of the collection. Users with less than 500 Dutch, self-written tweets were discarded. From the remaining users, 1000 users were randomly selected per class for further use in our research. More details about the processing of the users for use in the collection will follow in the section on *processing*.

3.2 ANNOTATION

In order to divide the users into income classes, they need to be annotated with their average yearly income. Distant supervision is used to find the average yearly income of a user. We look for an occupational title of a user in the user's biography line. In case a user has multiple occupations, we use the first one we can find. With the found title, we look up the user's occupational class and the average hourly income for that class. We then label the user with the average yearly income by multiplying the average hourly income with the average total hours worked in The Netherlands.

There are three additional data sources needed in order to make our annotation process work.

First, we use a list of occupational titles and their respective classes from Statistics Netherlands (2014a) to look up the occupation of a user. These

classes correspond with classes in the International Standard Classification of Occupations (International Labor Office, 2013), just like the earlier mentioned SOC. As this file was never meant for machinal consumption, the file had to be modified. All titles formed one long string, which had to be split in order to get the individual titles per class. Furthermore, the titles contained a lot of shorthand notations, e.g. "assistent-, coach" for the similar occupations "coach" and "assistent-coach" and slashes for synonyms like "typist / tekstverwerker". These were removed by hand as there was no suitable way to do this correctly in an automated manner. Second, we use a list of occupational classes and their respective average incomes from Statistics Netherlands (2014b) to look up the average hourly income for a particular class. We use the two-digit classes, as the incomes for almost all of them is known[1]. For most three- and four-digit classes, incomes aren't provided by Statistics Netherlands. Furthermore, the incomes among the two-digit groups varies enough in order to create a viable two-class split of our data. Finally, to derive the average yearly income we need to know the average worked hours per year in The Netherlands. According to the European Observatory of Working Life (2015) the average Dutch worker makes 1677 hours a year.

To evaluate the performance of our distant supervision method a random survey of 100 users per class was taken. Their labels were manually checked in a two class setting as mentioned in the previous subsection. The labels were considered correct if they appeared in the biography of a user, the user was a human and the occupational title was used to indicate paying occupation, not a hobby or study. The accuracy over the whole group of 200 users was 68,5 percent, with 59 percent in the low class and 78 percent in the high class. In 33 cases, the labels were wrong because the account was simply not used by a person but by a company. As there is no surefire way to distinguish between human and non-human users, we disregard these cases. The overall accuracy without these cases is 82 percent. The last few cases that were wrongly labeled consist of hobbies or voluntary work labelled as an occupation (15 cases), occupational titles that got bodged during the transformation of the needed file (11 cases) and four miscellaneous cases, which consist of users describing their former occupation in a non-trivially detectable way.

These results confirm that our distant supervision method yields enough correctly labeled data for our research, even though it is far from perfect. Possible future improvements of the method will follow in the *Discussion* section.

3.3 PROCESSING

Concerning the users, after extracting users with a known occupation from the twitter2 corpus a number of processing steps are involved to bring the entire group of 36113 users to a manageable number of suitable users. First, users that no longer exist, have their profile set as private or have less than 1000 tweets are removed to ensure we can get enough data per user for our research. The users are checked by using the Lookup API of Twitter, which allows us to check users in batches of 100. The remaining users are saved in a Python dictionary and written to disk with the built-in Pickle library.

[1] There were no average incomes available for the two-digit (submajor) groups 62, 82, 92 and 95. They are therefore left out of the rest of our research.

Afterwards, the remaining users are divided in the chosen income classes. From each class 1500 users are randomly selected for further processing by using the random.choice function of the NumPy Python library (van der Walt et al., 2011). For these users, their latest 1000 tweets are collected. Retweets and non-Dutch tweets were discarded. The language classification of each tweet was performed by the langid Python library (Lui and Baldwin, 2012). Users with less than 500 suitable tweets were left out of the data set. For all remaining users the tweets are written to a text file per user per class. From the remaining set of users, 1000 were randomly selected per group to be used in our research.

After the processing of the users is completed, the tweets are prepared for further use. URL's, hashtags and usernames are removed and the tweets are tokenized so that relevant features can be derived from them. The processing relies on the *TweetTokenizer* included in the NLTK Python library (Bird et al., 2009), a popular library for natural language processing in Python. As some features need to be generated from text tokenized per sentence, all tweets were run through a pre-trained NLTK sentence tokenizer created by Kiss and Strunk (2006). It was trained on the Dutch part of the Multilingual Corpus 1 (ECI), particularly on articles from the "De Limburger" newspaper.The resulting data was collected in a dictionary and written to disk with the Pickle library.

4 METHOD

Flekova et al. (2016) was a major source of inspiration for our methodology. However, as our research is limited in available time (7 weeks) and resources, we had to make some compromises in keep it workable, given all constraints. We mainly rely on the scikit-learn library (Pedregosa et al., 2011) for the concrete implementation of our method. It provides a lot of tools for machine learning, from full blown classifiers and regression models to tools for evaluation.

4.1 FEATURES

Originally, we planned to implement all of the feature groups used in Flekova et al. (2016). Due to the constraints of our research, we could only adopt two of the four groups: surface and readability. To compensate for this, we added a third groups of features: n-grams. All groups will be discussed in more detail below.

4.1.1 Surface

From this group, we chose and implemented the following features:

- Length of a user's tweets in words
- Length of a user's tweets in characters
- Average word length in a user's tweets
- Ratio of words longer than 5 characters in a user's tweets
- Type-token ratio

As these features mostly rely on counting words and performing some basic calculations, they were relatively easy to implement. The last two features partly overlap with the group of readability features as they are considered predictors of readability (Flekova et al., 2016).

4.1.2 Readability

Flekova et al. (2016) used a host of readability measures. They all have some commonality in the way they are calculated, but differ in measuring scale and intended application. Instead of implementing each measure with all its peculiarities by hand, we relied on the readability library (van Cranenburgh, 2016) for their calculation. This library provides a function that takes sentence tokenized text as its input and outputs the scores for several readability metrics. As sentence tokenization has already been performed by NLTK, we only need to provide the right input data for each user to calculate its scores.

The readability metrics included in our research are:

- Automated Readability Index
- Coleman-Liau Index
- Flesch-Kincaid Grade Level
- Flesch Reading Ease
- Gunning-Fog Index
- LIX Index
- SMOG Index

4.1.3 N-grams

For the final feature set, we looked at the syntax and style sets in the study by Flekova et al. (2016). The syntax set proved unimplementable in this short timeframe. Although the Alpino parser (Bouma et al., 2001) could provide us with the needed part-of-speech tags, it could not do so reliably. The parser would sometimes crash on input it could not handle.

We therefore moved on to the style set. Whilst some features like the ratio of words with numbers were easily implementable, others would prove more problematic. A lot of features depend on part-of-speech tags. Without a proper functioning parser we cannot implement those features at this moment. We had no other choice than looking elsewhere for a possible feature set to add.

Inspired by the work of fellow student Reinard van Dalen on classifying users based on their political preference, we decided to opt for word n-grams as our third and final feature set. We use a modified version of a function provided by our supervisor in order to generate the n-grams. It calls on NLTK's *ngrams* function to compute all possible n-grams for a given text.

We chose to include unigrams, bigrams and trigrams in our research. They are counted in a binary fashion: the fact that a certain n-gram occurs in a user's text is recorded, but the amount of occurrences is not taken into account.

4.2 CLASSIFICATION

As stated in the previous chapter, incomes are not available for every occupation, but only for a select number of occupational classes. The machine learning task ahead of us should therefore be framed as classification, instead of regression. Handling in the spirit of Flekova et al., we tried to find a classification method that is closest to the methods they have used, despite the difference in prediction task. They used linear regression and vector support regression with an RBF kernel as its non-linear counterpart for comparison.

Two classification methods came to mind: a support vector machine (SVM) and logistic regression. We chose to only use the latter. Scikit-learn's documentation on SVM's states that "if the number of features is much greater than the number of samples, the method is likely to give poor performance" (scikit-learn, 2010). As one of our feature sets is n-grams, one can expect thousands of features while the data set only consists of two

thousand samples. We therefore shifted our focus to logistic regression and implemented it in our classifier using scikit-learn's *LogisticRegression* module.

4.3 EVALUATION

After implementing all features, the performance of the classifier and its composing features is assessed by using k-fold cross validation. In our case we apply a k of 10, like Flekova et al. (2016).

We rely on scikit-learn's *KFold* module to split our data for each of the ten folds. For each fold, the results were printed to screen, including a list of most informative features. After each validation run the precision, recall and F1-scores are calculated in order to compare the different classifier setups. We test a host of different setups in order to find the most effective combination (i.e. giving us the highest F1-score with the least amount of features). The tested feature compositions are shown in table 1.

The classifiers will all be compared to a baseline. As the data set consists of two equally large groups, a majority baseline cannot be used. We therefore use a random baseline. Given that there are two classes and they are equal in size, the chance that a single instance belongs to a certain class is 50 percent. From this follows that precision (P), recall (R) and the harmonic mean of both of them (F1-score) will all be 0.5.

#	Set 1	Set 2	Set 3
1	Surface	Readability	N-grams (n=1-2-3)
2	Surface	Readability	N-grams (n=2-3)
3	Surface	Readability	N-grams (n=1-3)
4	Surface	Readability	N-grams (n=1-2)
5	Surface	Readability	N-grams (n=3)
6	Surface	Readability	N-grams (n=2)
7	Surface	Readability	N-grams (n=1)
8	Surface	Readability	-
9	Surface	-	N-grams (n=1-2-3)
10	Surface		N-grams (n=2-3)
11	Surface	-	N-grams (n=1-3)
12	Surface	-	N-grams (n=1-2)
13	Surface	-	N-grams (n=3)
14	Surface	-	N-grams (n=2)
15	Surface	-	N-grams (n=1)
16	Surface	-	-
17	-	Readability	N-grams (n=1-2-3)
18	-	Readability	N-grams (n=2-3)
19	-	Readability	N-grams (n=1-3)
20	-	Readability	N-grams (n=1-2)
21	-	Readability	N-grams (n=3)
22	-	Readability	N-grams (n=2)
23	-	Readability	N-grams (n=1)
24	-	Readability	-
25	-	-	N-grams (n=1-2-3)
26			N-grams (n=2-3)
27			N-grams (n=1-3)
28			N-grams (n=1-2)
29			N-grams (n=3)
30			N-grams (n=2)
31			N-grams (n=1)

Table 1: An overview of the classifier setups included in our evaluation.

5 | RESULTS AND DISCUSSION

To be written on Tuesday.

6 CONCLUSION

To be written on Tuesday.

BIBLIOGRAPHY

Bird, S., E. Klein, and E. Loper (2009). *Natural Language Processing with Python*. O'Reilly Media.

Bouma, G., G. Van Noord, and R. Malouf (2001). Alpino: Wide-coverage computational analysis of dutch. *Language and Computers 37*(1), 45–59.

European Observatory of Working Life (2015). *Developments in collectively agreed working time 2014*.

Flekova, L., D. Preoţiuc-Pietro, and L. Ungar (2016, August). Exploring stylistic variation with age and income on twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, pp. 313–319. Association for Computational Linguistics.

International Labor Office (2013). *International Standard Classification of Occupations 2008 (ISCO-08)*. International Labor Office.

Kiss, T. and J. Strunk (2006, 2016/12/18). Unsupervised multilingual sentence boundary detection. *Computational Linguistics 32*(4), 485–525.

Li, J., A. Ritter, and E. Hovy (2014, June). Weakly supervised user profile extraction from twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, pp. 165–174. Association for Computational Linguistics.

Lui, M. and T. Baldwin (2012). Langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, Stroudsburg, PA, USA, pp. 25–30. Association for Computational Linguistics.

Office for National Statistics (2010). *The Standard Occupational Classification (SOC) 2010 Vol 1: Structure and Descriptions of Unit Groups*. Palgrave Macmillan.

Pedregosa, F., 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.

Preotiuc-Pietro, D., V. Lampos, and N. Aletras (2015, July). An analysis of the user occupational class through twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China, pp. 1754–1764. Association for Computational Linguistics.

scikit-learn (2010). 1.4. support vector machines.

Sloan, L., J. Morgan, P. Burnap, and M. Williams (2015, 03). Who tweets? deriving the demographic characteristics of age, occupation and social class from twitter user meta-data. *PLoS ONE 10*(3), 1–20.

Statistics Netherlands (2014a). *Codelijsten ISCO-08*. Retrieved from https://www.cbs.nl/nl-nl/onze-diensten/methoden/classificaties/onderwijs

Statistics Netherlands (2014b). *Uurlonen van werknemers naar beroepsgroep, 2012*. Retrieved from https://www.cbs.nl/nl-nl/maatwerk/2014/15/uurlonen-van-werknemers-naar-beroepsgroep-2012.

van Cranenburgh, A. (2016). Readability.

van der Walt, S., S. C. Colbert, and G. Varoquaux (2011, March). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering 13*(2), 22–30.