Predicting the Age of Social Network Users from User-Generated Texts with Word Embeddings

Anton Alekseev
Steklov Mathematical Institute at St. Petersburg
St. Petersburg, Russia
anton.m.alexeyev@gmail.com

Sergey I. Nikolenko
Steklov Mathematical Institute at St. Petersburg, Russia
Kazan Federal University, Kazan, Russia
sergey@logic.pdmi.ras.ru

***Abstract*—Many web-based applications such as advertising or recommender systems often critically depend on the demographic information, which may be unavailable for new or anonymous users. We study the problem of predicting demographic information based on user-generated texts on a Russian-language dataset from a large social network. We evaluate the efficiency of age prediction algorithms based on *word2vec* word embeddings and conduct a comprehensive experimental evaluation, comparing these algorithms with each other and with classical baseline approaches.**

I. Introduction

The recent development of deep learning techniques for natural language processing has led to state of the art models that operate in a basically unsupervised fashion and do not require much linguistic insight; one such direction of study deals with *word embeddings*, vector representations of words that capture certain semantic relations between the words and can serve as an intermediate step for subsequent models.

In this work, we concentrate on a novel application of word embeddings to *user profiling*, focusing on improving user age prediction with full-text items. We believe that huge corpora of user-generated texts stored in forums and social networks can be used to produce interpretable, semantic user profiles and improve recommendations for full-text items. In this work, we develop new age prediction methods and algorithms for users interacting with full-text items based on distributed word representations.

User profiling by user behaviour has had a long history in many different contexts, but text-based user profiling has not attracted too much attention. Previous attempts at big data user profiling without deep neural networks have leaned upon external knowledge in the form of ontologies [41] and presented a general framework for using NLP in profiling [10]. There is a large classical field of authorship analysis, attribution and author verification studies [31], [74]; since this is not really our focus, we refer to surveys [12], [65], [66] for details and references.

One could, however, find some works that use natural language processing to perform or augment user profiling. In particular, there have been several works closer to social sciences based on available anonymized datasets that do things similar to user profiling, usually mining demographic information from texts generated by a user, and attempts to mine text to establish new information regarding a user or relations between users. If it is done from social network data, the field is known as *social media personal analytics*. Next, we highlight some of this research. In [35], anonymized text messaging datasets are used to investigate the demographics of texting, while in [21], author profiling for English emails uncovers basic demographic traits (gender, age, geographic origin, level of education, and native language) and five psychometric traits based on email texts. Several Twitter-based studies have focused on mining demographic features based on tweets [19], [26]; the work [34], for instance, does it in a weakly supervised fashion, using Facebook or Google+ profiles as distant supervision. The work [49] detects personality traits from weblog texts, while the work [5] explicitly studies lexical predictors of personality type, [9] determines demographic information by social media texts, and [55] mines user relations from online discussions; an interesting extension is [22] which attempts personality profiling of fictional characters based on the texts about them. In [58], author profiles in social media are mined to get hidden user profile information, while in [50] metadata is used to mine author profiles; the work [70] attempts automatic collection and summarization of personal profiles from various social networks and other sources, while [17] proposes linguistic features that help determine the natural language of a person writing in English (on a dataset of the First NLI Shared Task) and [54] determines a user's occupation by his or her tweets. In [18], [69], the user's political preferences are determined by his or her tweets, and [32] drives it further to get the user's actual voting intentions. This kind of profiling even extends to medical issues: the work [52] attempts to screen Twitter users for depression based on their tweets. Numerous works on the topic have been published based on the results of the shared Author Profiling Tasks at digital text forensics events by PAN initiative [23], [59]–[62]. Finally, there are quite a few works for determining the geographical location of a user from his or her textual activity in social networks [8], [27], [36], [56], [57], [71].

As for neural NLP models, one recent work that actually uses modern neural network-based NLP to automatically construct user profiles is [67]. There, convolutional neural networks are used to construct a joint representation of users, products, and their reviews, in particular user profiles. This results in semantic user profiles that are then used to improve sentiment classification but can probably be used for other purposes as well. A recent work [48] has used word embeddings to construct user profiles from the texts they liked in a social network; the profiles were constructed as logistic regression weights of word clusters (clustered in the semantic space of word embeddings), with a special

ISBN 978-952-68397-8-3 (PDF)

mechanism to reduce the weights of clusters with common words and bring topical clusters to the top. In [1], a *deep semantic similarity model* (DSSM) is trained to model the "interestingness" of documents. The purpose of the model is to recommend target documents that might interest a user based on a source document which she is reading at the moment. This is mostly an information retrieval model, trained on click transitions between source and target documents; this work is similar to [68] and also uses convolutional architectures. The hierarchical neural language model from [20] with a document level and a token level can also be extended to learning user-specific vectors to represent individual preferences, which can be used to give personalized recommendations.

In this work, we propose several relatively simple algorithms that operate on word embeddings of the words in social network statuses of the users, aiming to predict a user's age from his or her writing. The paper is organized as follows. In Section II, we describe the dataset used in this work. Section III discusses word embeddings, one of the most important tools in modern natural language processing. Section IV describes in detail the age prediction algorithms that we propose and evaluate in this work. Section V is devoted to a comprehensive experimental study, where we evaluate and compare not only the proposed algorithms with each other but also *word2vec* models trained with different parameters; our aim here is to draw practically important conclusions that may be useful for subsequent studies of the Russian language. Finally, Section VI concludes the paper.

II. Problem and dataset

For this project, we have obtained a large dataset from the *Odnoklassniki* social network. The dataset has been created as follows:

(1) the dataset began with 486 seed users;
(2) for these users, their sets of friends have been extracted;
(3) then the friends of these friends; as a result, the dataset contains a neighborhood of depth 2 in the social graph for the original seed users.

As a result, the dataset contains information on 868,126 users of the *Odnoklassniki* social network. In particular, it contains the following data:

(1) demographic information on 868,126 users of the network: gender, age, and region (region info may be imprecise since there is no such explicit field in the user's profile, the region is determined by the IP addresses from which the user has logged in most often);
(2) the social graph that defines the "friendship" relation and contains (and indicates) several different type of links: "friend", "love", "spouse", "parent", and so on; all users with known demographic data are also present in the social graph;
(3) history of logins for individual users;
(4) data on the "likes" ("class" marks) a user has given to other users' statuses and posts in various groups;
(5) texts of the posts for individual users and group statuses that have been liked by these selected users.

The mean age of all users was 31.39 years; the age distribution is shown on Fig. 1. It is important to note that there are quite a lot of users with implausible ages (ages 2 and 3, age higher than 100 years); since the user specifies the age by himself/herself, this probably represents missing, incorrect, or purposefully distorted data. Note that this is an important point for the relevance of our research: when a user has not specified his/her age, or when a user has specified an obviously incorrect age, we still need to be able to predict his or her age in order to give age-related recommendations and enroll the user into age cohorts.

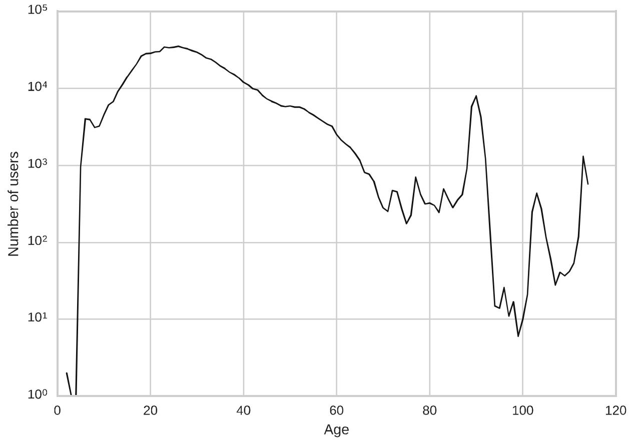

Fig. 1. Age distribution in the *Odnoklassniki* dataset

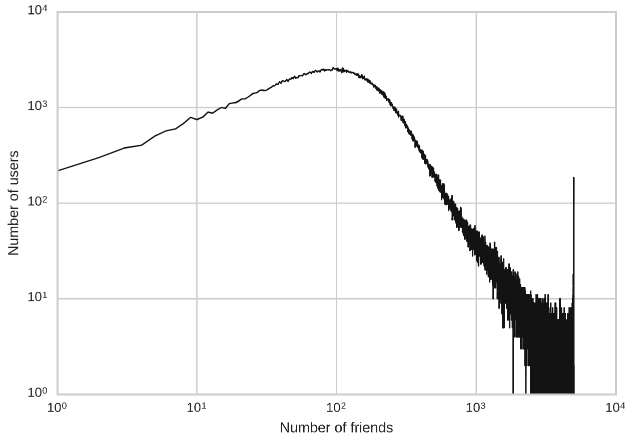

Fig. 2. Number of friends distribution in the *Odnoklassniki* dataset

For the experiments, however, we have removed from the dataset all ages below 10 and above 80 since they are likely to correspond to faulty/missing information.

Fig. 2 shows the distribution of the number of friends in the *Odnoklassniki* dataset; interestingly, while the usual Pareto distribution (straight line on a log-log plot) picks up after about 100 friends, it actually increases before that point. This is probably an artifact of the data collection: naturally, the social circle (neighborhood of depth 2) of a predefined set of seed users will contain few isolated or nearly isolated users.

III. Word embeddings

A. Background

Recent advances in distributed word representations have made it into a method of choice for modern natural language processing [24]. Distributed word representations are models that map each word occurring in the dictionary to a Euclidean space, attempting to capture semantic relationships between the words as geometric relationships in the Euclidean space. In a classical word embedding model, one first constructs a vocabulary with one-hot representations of individual words, where each word corresponds to its own dimension, and then trains representations for individual words starting from there, basically as a dimensionality reduction problem. For this purpose, researchers have usually employed a model with one hidden layer that attempts to predict the next word based on a window of several preceding words. Then representations learned at the hidden layer are taken to be the word's features.

The modern field of word embeddings started with the work [6], subsequently extended in [7]. Extending previous work on statistical language models that were usually based on word n-grams [13], [15], [25], [30], Bengio et al. proposed the idea of *distributed word representations* that operate as follows:

(1) for each vocabulary word $w \in V$, associate it with a feature vector $C(w)$ (word embedding) $v_w \in \mathbb{R}^d$ (typical values of d lie in the hundreds);

(2) express the probability function of words appearing in context windows via these vectors as

$$g(i, C(w_{t-1}, ..., C(w_{t-n+1})) = \hat{P}(w_t|w_1^{t-1})$$

where $C(w_{t-1}, ..., C(w_{t-n+1}$ are vectors of context words and g is a parameterized function with parameters ω;

(3) train from a large unlabeled text corpus both the vectors and the parameters of this probability function; the objective maximized during training is the corpus log-likelihood

$$L = \frac{1}{T}\sum_t \log f(w_t, w_{t-1}, ..., w_{t-n+1}; \Theta) + R(\Theta)$$

where $\Theta = (C, \omega)$, and $R(\Theta)$ is a regularization term.

There exist two most commonly used models for word embeddings, both introduced in [42]: *Continuous Bag-of-Words* (CBOW) and *skip-gram*. During its learning, a CBOW model is trying to reconstruct the words from their contexts. It is done by a network whose architecture is shown on Fig. 3a; the training process for this model proceeds as follows:

(1) each of the inputs of this network is a one-hot encoded vector of size $|V|$, where V is the vocabulary;

(2) when computing the output of the hidden layer, we take an average of all input vectors; the hidden layer is basically a matrix of vector embeddings of words, so the nth row represents an embedding of the nth word in the vocabulary;

(3) the output layer represents a score u_j for each word in the vocabulary; to obtain the posterior, which is a multinomial distribution, we then use the softmax

$$\hat{P}(w_t|w_1^{t-1}) = \frac{\exp(u_j)}{\sum_{j'=1}^{t}\exp(u_{j'})},$$

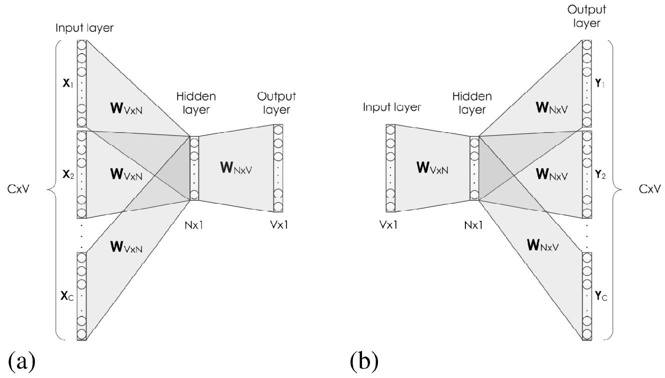

Fig. 3. *word2vec* network architectures (pictures modeled after [64]): (a) CBOW; (b) skip-gram

so the loss function is

$$E = -\log p(w_t|w_1^{t-1}) = -u_j + \log\sum_{j'=1}^{|V|}\exp(u_{j'}).$$

The skip-gram model operates in a somewhat inverse manner, which can be seen from its network architecture shown on Fig. 3b. Here the target is an input word, and the output layer, in turn, now represents C multinomial disctibutions

$$\hat{P}(w_1^{t-1}|w_t) = \frac{\exp(u_{cj})}{\sum_{j'=1}^{i}\exp(u_{j'})}$$

with the loss computed as

$$E = -\log p(w_1^{t-1}|w_t) = -\sum_{c=1}^{C} u_{jc} + C\log\sum_{j'=1}^{|V|}\exp(u_{j'})$$

The idea of word embeddings has been applied back to language modeling, e.g., in [43], [44], [46], and then, starting from the works of Mikolov et al. [42], [45], word representations have been applied for numerous natural language processing problems, including text classification, extraction of sentiment lexicons, part-of-speech tagging, syntactic parsing and so on. Basically all models that we review below either make use of one of the word embedding models or construct character-level embeddings.

Another important model for word embeddings is *Glove* (GLObal VEctors for word representations) [53]. In the Glove model, the objective function for training word embeddings w_i and $\tilde{w}_i$ is

$$J = \sum_{i,j=1}^{V} f(X_{ij})\left(\boldsymbol{w}_i^\top \tilde{\boldsymbol{w}}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2,$$

where $X \in \mathbb{R}^{V\times V}$ is the cooccurrence matrix between words, so X_{ij} is the frequency of word i cooccurring with word j, $X_i = \sum_j X_{ij}$ is the total number of occurrences for word i, $\boldsymbol{w} \in \mathbb{R}^d$ is the word embedding (in dimension d), and $\tilde{\boldsymbol{w}} \in \mathbb{R}^d$ is the context word embedding in dimension d, and f is a function that does not overweigh frequent cooccurrences too

TABLE I. WORD2VEC MODELS TRAINED ON LARGE RUSSIAN CORPORA. THE COLUMNS ARE: DIMENSION d, WINDOW SIZE w, NUMBER OF NEGATIVE SAMPLES n, VOCABULARY THRESHOLD v, AND THE RESULTING MODEL SIZE.

Type	d	w	n	v	Size
CBOW	100	11	10	20	1.3G
CBOW	100	11	10	30	0.97G
skip-gram	100	11	10	30	0.97G
skip-gram	200	11	1	20	1.3G
CBOW	200	11	1	30	2.0G
skip-gram	200	11	1	30	2.0G
CBOW	300	11	1	30	2.9G
skip-gram	300	11	1	30	2.9G

much; usually Glove employs the weights

$$f(x) = \begin{cases} \left(\frac{x}{x_{\max}}\right)^{\alpha}, & \text{if } x < x_{\max}, \\ 1, & \text{otherwise.} \end{cases}$$

The idea is to express $P_{ij} = \frac{X_{ij}}{X_i}$, the probabilities that word j occurs in the context of word i, and natural requirements on the objective function (e.g., the fact that after transposing the X matrix we should replace w_i with $\tilde{w}_i$ and vice versa) lead to this optimization problem. Pennington et al. report improved results for named entity recognition [53], and since then Glove vectors have been used for a number of different NLP tasks.

Efficient and/or more stable algorithms for training word embeddings have been developed in [39], [40], [42], [47].

B. Word2vec models

As a dataset for word embeddings, we have used a large Russian-language corpus (the largest we know) with about 14G tokens in 2.5M documents [4], [51]. This corpus includes:

- Russian *Wikipedia*: 1.15M documents, 238M tokens;
- automated Web crawl data: 890K documents, 568M tokens;
- (main part) the huge *lib.rus.ec* library corpus: 234K documents, 12.9G tokens;
- user statuses and group posts from the *Odnoklassniki* social network, as described above.

All of this has let us obtain what we believe to be an unprecedented quality of the resulting representations. We refer to [4], [51] for more details on the training data.

We have used continuous bag-of-words (CBOW) and skip n-gram *word2vec* models trained on a single NVidia Titan X GPU with the currently fastest *word2vec* implementation ported to CUDA(https://github.com/ChenglongChen/word2vec_cbow). Our previous experiments have suggested that vector sizes in the low hundreds and window size of 11 words are the best parameters on this dataset. In total, so far in the experiments we have used eight different *word2vec* models whose parameters are shown in Table I; the models differ in the type (CBOW or skip-gram), dimension of word vectors d, window size w (later omitted since $w = 11$ in all models), number of negative samples n in the training, and vocabulary threshold v that controls the size of the vocabulary (a lower threshold means more words get vectors, but words with few occurrences will not have enough training data and might have a random-like, meaningless vector). Note also that every model can come in a “raw” form, as trained, and a normalized form where all vectors are normalized to Euclidean length 1.

IV. USER PROFILING ALGORITHMS

A. Mean age of friends

In this section, we show the basic user profiling algorithms that we have used to predict a user’s age based on the texts of the user’s statuses.

First, folklore among social network researchers says that to predict a user’s age it is usually sufficient to take the mean age of his or her friends: it will predict the age with outstanding accuracy. We have tested this theory on the OK dataset. To investigate, we have trained the following models:

(1) MEANAGE: predict age with the mean of friends’ ages and the global mean if no friends ages are known;
(2) LINEARREGR: linear regression with a single feature (mean friends age);
(3) ELASTICNET: elastic net regressor with a single feature (mean friends age);
(4) GRADBOOST: gradient boosting with a single feature (mean friends age).

Results of these simple models are shown in Table II in two variations: basic, where we substitute zeros instead of missing features (when there are no friends’ ages) and “nonzero”, where we train and test only on a subset of data with nonzero features (at least one friend with known age). It appears that LINEARREGR performs worse than MEANAGE in its first variation because linear regression cannot implement the condition “if the feature is zero (default value in the absence of neighbors) do something completely different”, and GRADBOOST is noticeably better because it is powerful enough to handle such case-by-case conditions.

However, we should note that the errors here are quite significant: in terms of MAE, we are more than nine years off on average even if we restrict ourselves to cases with friends with known ages. Hence, we expect that subsequent work is not meaningless and can bring substantial improvements.

B. Algorithms based on word embeddings

Note that while the idea to use the sum and/or mean of word embeddings to represent a sentence/paragraph is, indeed, the simplest idea for the representation of a larger chunk of text, due to the geometric properties of the *word2vec* and GloVe models this idea is not as naive as it sounds. This approach has been used as a baseline in [33] but was proposed as a reasonable method for short phrases in [45] and has been shown to be effective for document summarization in [29].

Thus, we propose three basic algorithms:

(1) MEANVEC: train on mean vectors of all statuses for a user;
(2) LARGESTCLUSTER: train on the centroid of the largest cluster of statuses;

TABLE II. Baseline results: predictions by mean age of the friends

Model	Train		Test		Train, nonzero		Test, nonzero	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
MeanAge	9.701	6.725	9.661	6.707	8.833	5.865	8.787	5.840
LinearRegr	11.252	8.659	11.226	8.650	8.794	5.951	8.752	5.930
ElasticNet	11.251	8.660	11.226	8.651	8.795	5.969	8.752	5.948
GradBoost	9.602	6.743	9.569	6.730	8.683	5.879	8.645	5.861

(3) AllMeanV: train on every status independently, with the mean vector of a specific status and the mean age of friends as features and the user's demography as target; at the testing stage, we compute predictions for every status and average the predictions.

The MeanVec algorithm simply computes the mean vector of all statuses and adds it as features to the classification/regression model. Formally speaking, we introduce the following notation:

- W is the vocabulary, with words $w \in W$;
- U is the set of users, a user will usually be denoted as $u \in U$;
- S_u is the set of texts "belonging to" user u (either written by u or liked by him/her), with a single text usually denoted as $s \in S_u$; the s stands for either "string" or, more specifically, "status";
- $\mathbf{v}_w^m$ is the vector (word embedding) of word w in model m (we will omit the superscript when it is not important or clear from context);
- $\bar{\mathbf{v}}_A = \frac{1}{|A|}\sum_{w\in A}\mathbf{v}_w$ is the mean vector of a set of word embeddings A;
- MAF_u is the mean age of the friends of a user $u \in U$; in the algorithms, this is the only feature we use from the social graph.

In this notation, the MeanVec algorithm operates as follows: for a machine learning (regression for age) algorithm ML,

```
(1) for every user $u \in U$:
      • for every status $s \in S_u$, compute its mean vector $\bar{\mathbf{v}}_s = \frac{1}{|s|}\sum_{w\in S}\mathbf{v}_w$;
      • compute the mean vector of all statuses $\bar{\mathbf{v}}_u = \frac{1}{|S_u|}\sum_{s\in S_u}\bar{\mathbf{v}}_s$;
(2) train ML with features $(\mathrm{MAF}_u, \bar{\mathbf{v}}_u)$ for every $u \in U$.
```

The LargestCluster algorithm operates as follows: for a machine learning algorithm ML,

```
(1) for every user $u \in U$:
      • for every status $s \in S_u$, compute its mean vector $\bar{\mathbf{v}}_s = \frac{1}{|s|}\sum_{w\in S}\mathbf{v}_w$;
      • cluster the set of vectors $\{\bar{\mathbf{v}}_s \mid s \in S\}$ into two clusters with agglomerative clustering; denote by $C \subseteq S$ the larger cluster;
      • compute the mean vector of statuses from $C$ $\bar{\mathbf{c}}_u = \frac{1}{|C|}\sum_{s\in C}\bar{\mathbf{v}}_s$;
(2) train ML with features $(\mathrm{MAF}_u, \bar{\mathbf{c}}_u)$ for every $u \in U$.
```

TABLE III. Basic statistics for extended and basic datasets

Dataset	Training set		Test set	
	Users	Statuses	Users	Statuses
Extended	661206	10880321	165301	2704883
Basic	170856	2014983	42713	503150

The AllMeanV algorithm operates as follows: for a machine learning algorithm ML,

```
(1) for every user $u \in U$ and every status $s \in S_u$, compute its mean vector $\bar{\mathbf{v}}_s = \frac{1}{|s|}\sum_{w\in S}\mathbf{v}_w$;
(2) train ML with features $(\mathrm{MAF}_u, \bar{\mathbf{v}}_s)$ for every $u \in U$ and $s \in S_u$;
(3) on the prediction stage, for a user $u \in U_{\mathrm{test}}$:
      • for every status $s \in S_u$, compute its mean vector $\bar{\mathbf{v}}_s = \frac{1}{|s|}\sum_{w\in S}\mathbf{v}_w$;
      • predict the age for this status, $a_s = \mathrm{ML}(\mathrm{MAF}_u, \bar{\mathbf{v}}_s)$;
      • return the average predicted age, $a = \frac{1}{|S_u|}\sum_{s\in S_u}\mathrm{ML}(\mathrm{MAF}_u, \bar{\mathbf{v}}_s)$.
```

V. Experimental evaluation

A. Setting

We began evaluation with the entire dataset as outlined above, what is called below the "extended" dataset. However, in order to perform more experiments, be more flexible, and not get bogged down in the technicalities of fitting huge datasets into available hardware, we have also prepared a smaller "basic" dataset that we performed some experiments on. The basic dataset preserves most properties of the extended dataset; the only difference is that we have filtered the users to have at least 5 and at most 300 statuses. This has let us cut off a relatively small number of highly prolific writers (or, to be more precise, prolific reposters), significantly reducing the total number of statuses, and cut off the long tail of users with very few statuses, while still preserving important properties of the data. The basic statistics for the two datasets are shown on Table III, and Fig. 4 indicates that all basic distributions such as age and number of friends are very similar for the two datasets, except, naturally, the distribution of the number of statuses. Both datasets were splitted into training and test sets randomly using 80 and 20 percentages, respectively.

B. *Comparing* word2vec *models*

In the first experiment, we took the simplest MeanVec algorithm and compared how various *word2vec* models perform. The results are shown in Table IV. We can draw the following conclusions:

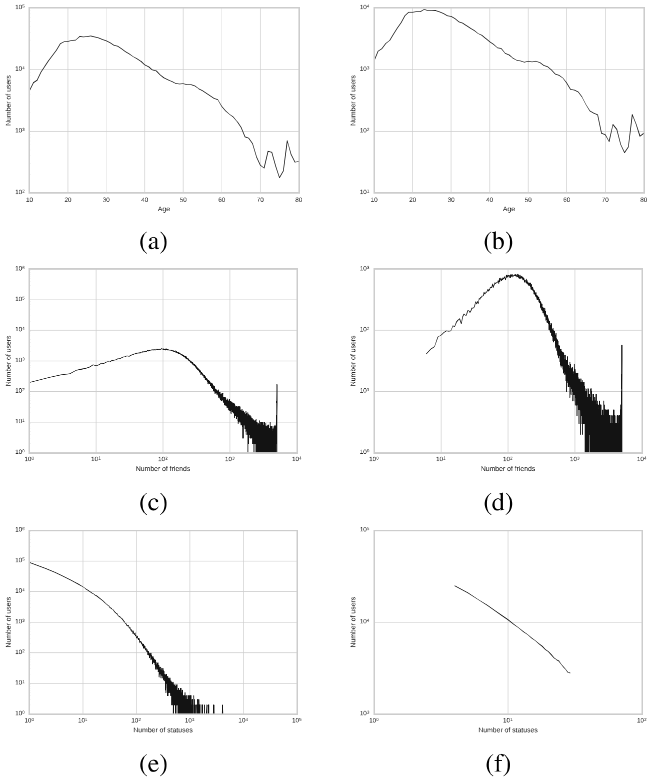

Fig. 4. Basic distributions for the extended and basic datasets

- naturally, the MEANAGE algorithm does not care about *word2vec* at all, it is only included as a sanity check;
- *word2vec* models do help all models, both linear and GRADBOOST – compare these results with Table II;
- it appears that CBOW models outperform skip-gram models in this task (quite significantly);
- by increasing the dimension d, we also get some improvements, but these improvements are rather small;
- a decrease in v, although it makes the *word2vec* model significantly larger and longer to train, has absolutely no effect on the end result.

Generally speaking, these conclusions mean that for the purposes of demographic analysis and similar problems we can concentrate on relatively small *word2vec* models, with dimensions 100 or 200, and perhaps further increase v, which would lead to much smaller models and faster training.

In the second experiment here, we have compared raw and normalized *word2vec* models in the same setting; some of the results are shown in Table V; for convenience, raw and normalized versions are shown immediately next to each other. The results are rather interesting: the more expressive is the classifier, the better normalized versions are. For LINEARREGR, raw vectors slightly outperform normalized ones, for ELASTICNET there is almost no difference, and GRADBOOST makes (sometimes significantly) better use of the normalized versions. This result can probably be attributed to the fact that while normalized vectors are indeed usually recommended for use, raw vectors can have larger absolute values, including rather large outliers, and simple linear models are better at picking on larger absolute values. Still, the conclusion is to use mostly normalized models in the future since we are after the best model rather than the best linear regression.

C. Comparing MEANVEC *and* LARGESTCLUSTER

The next step was to compare baseline algorithms with each other. Table VI shows the comparison results between MEANVEC and LARGESTCLUSTER algorithms (marked MV and LC) on the original (extended) dataset, shown for a selection of normalized *word2vec* models.

Interestingly, the LARGESTCLUSTER algorithm invariably loses to MEANVEC in all experiments. One possible reason for this might be that the largest cluster of all statuses turns out in many cases to be the least meaningful (e.g., consisting of similar reposts from an online game or of extremely brief statuses, e.g., consisting of a single smiley); we have verified this idea with a direct examination of the data but believe that in the future, variations on the idea of clustering statuses might yet prove to be useful.

D. Comparing MEANVEC, LARGESTCLUSTER, *and* ALLMEANV

This comparison has been performed on the smaller "basic" dataset that we have presented above. Results of this comparison are shown in Table VII, which marks the MEANVEC, LARGESTCLUSTER, and ALLMEANV algorithms as MV, LC, and AV respectively.

As for the results, the LARGESTCLUSTER algorithm, again, loses in almost all cases to both MEANVEC and ALLMEANV. What is much more interesting, however, is that ALLMEANV, while performing roughly on par with MEANVEC in LINEARREGR and ELASTICNET, begins to lose significantly to MEANVEC and even LARGESTCLUSTER when we use GRADBOOST as the classifier. This result was quite surprising since we expected that more data and more detailed status vectors (individual for each status rather than averaged over all statuses of a user) will actually bring an improvement. One possible reason for this behaviour is that in passing from MEANVEC to ALLMEANV we have, in essence, "moved" the averaging from the semantic space of word embeddings to averaging prediction results. Hence, this result can be interpreted as showing that simple averaging works very well in the semantic space (this is not surprising given that many semantic relations become linear in the space of embeddings), even better than building an ensemble of predictions from individual statuses afterwards.

VI. CONCLUSION

In this work, we have prepared and preprocessed a huge Russian language free text dataset with a number of different sources ranging from literature to user statuses in social networks, trained a number of *word2vec* models, obtained and preprocessed a large user profiling dataset from the social network *Odnoklassniki*, suggested a number of user profiling algorithms based on *word2vec* embeddings, and performed a large-scale comparison of these algorithms and different *word2vec* models, drawing conclusions important for subsequent work on user-generated texts.

TABLE IV. A COMPARISON OF *word2vec* MODELS ON THE EXTENDED DATASET WITH THE MEANVEC ALGORITHM

Word2vec params				Train		Test		Train, nonzero		Test, nonzero	
type	d	n	v	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
MEANAGE											
				9.672	6.707	9.668	6.711	8.778	5.835	8.800	5.848
LINEARREGR											
cbow	100	10	20	10.401	7.776	10.397	7.785	8.514	5.792	8.541	5.810
cbow	100	10	30	10.402	7.777	10.396	7.784	8.515	5.791	8.539	5.808
skip	100	10	30	10.818	8.219	10.813	8.232	8.624	5.863	8.645	5.879
skip	200	1	20	10.738	8.146	10.726	8.151	8.593	5.847	8.613	5.859
cbow	200	1	30	10.355	7.737	10.349	7.743	8.497	5.782	8.520	5.798
skip	200	1	30	10.735	8.143	10.724	8.148	8.592	5.846	8.613	5.859
cbow	300	1	30	10.338	7.722	10.329	7.727	8.492	5.779	8.512	5.794
skip	300	1	30	10.689	8.088	10.675	8.096	8.583	5.837	8.601	5.854
ELASTICNET											
cbow	100	10	20	10.810	8.208	10.799	8.217	8.694	5.903	8.719	5.921
cbow	100	10	30	10.806	8.203	10.795	8.212	8.702	5.909	8.726	5.926
skip	100	10	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
skip	200	1	20	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
cbow	200	1	30	10.949	8.349	10.937	8.359	8.736	5.934	8.760	5.951
skip	200	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
cbow	300	1	30	11.026	8.433	11.017	8.445	8.741	5.938	8.766	5.956
skip	300	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
GRADBOOST											
cbow	100	10	20	9.089	6.352	9.065	6.344	8.399	5.697	8.394	5.699
cbow	100	10	30	9.093	6.356	9.066	6.345	8.401	5.699	8.395	5.700
skip	100	10	30	9.294	6.527	9.277	6.529	8.495	5.766	8.491	5.770
skip	200	1	20	9.363	6.580	9.342	6.576	8.519	5.785	8.512	5.785
cbow	200	1	30	9.067	6.341	9.043	6.333	8.383	5.682	8.377	5.683
skip	200	1	30	9.365	6.583	9.344	6.580	8.520	5.784	8.512	5.785
cbow	300	1	30	9.048	6.323	9.025	6.316	8.380	5.683	8.371	5.681
skip	300	1	30	9.387	6.596	9.367	6.595	8.536	5.799	8.532	5.804

While the proposed algorithms did bring certain improvements as compared to the “zero baseline” of training with the mean age of a user’s friends, these improvements were not huge in absolute terms: we have been able to shave off about 0.2 years in terms of mean absolute error. Therefore, we remain optimistic that these results can be much improved in the future. In further work, we plan to:

(1) develop new features for user profiling algorithms based on text embeddings (embedding larger portions of text than a word); here we hope to train a deep text understanding model for the Russian language and apply it to user profiling. Second, we plan to

(2) develop and train a character-level word embedding model for the Russian language; we expect this model to be very important for studies of user-generated texts since they abound with typos, intentional misspellings, variations and so on.

Also, apart from developing new user profiling algorithms, we plan to investigate other variations of word embeddings. For example, one such is given by the Polyglot system [3], and a completely different direction with a graph-based model is proposed in [2]. We also note recent efforts in *word sense disambiguation* for word embeddings: the same word can have several very different meanings, and it would be natural to try to model it with several vectors in the semantic space [11], [14], [16], [28], [37], [38], [63], [72], [73]. In further work, we plan to perform an even more extensive comparison between various word embedding variations; a comparison across these models might provide valuable insight into the use of *word2vec* models for subsequent applications such as user profiling, sentiment analysis, or full-text recommendations.

ACKNOWLEDGMENTS

This work was partially supported by the Russian Federation grant 14.Z50.31.0030 and by the 2016 grant “User Profiling Based on Distributed Word Representations” sponsored by Samsung Russia. We also thank Dmitry Bugaichenko, Philipp Fedchin, and the *Odnoklassniki* social network (from the *Mail.Ru* holding) for providing us with a large dataset of social network statuses in Russian.

REFERENCES

[1] *Modeling Interestingness with Deep Neural Networks*. EMNLP, 2014.

[2] C. Aggarwal and P. Zhao. Graphical models for text: A new paradigm for text representation and processing. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’10, pages 899–900, New York, NY, USA, 2010. ACM.

[3] R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

TABLE V. A comparison of *word2vec* models with their normalized versions on the extended dataset with the MeanVec algorithm

Word2vec params				Train		Test		Train, nonzero		Test, nonzero	
type	d	n	v	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
LinearRegr											
cbow	100	10	20	10.402	7.777	10.396	7.784	8.515	5.791	8.539	5.808
n.cbow	100	10	20	10.426	7.807	10.418	7.807	8.535	5.807	8.560	5.824
cbow	100	10	30	10.818	8.219	10.813	8.232	8.624	5.863	8.645	5.879
n.cbow	100	10	30	10.839	8.231	10.826	8.240	8.630	5.864	8.649	5.879
skip	100	10	30	10.738	8.146	10.726	8.151	8.593	5.847	8.613	5.859
n.skip	100	10	30	10.753	8.156	10.736	8.159	8.601	5.853	8.620	5.864
skip	200	1	20	10.355	7.737	10.349	7.743	8.497	5.782	8.520	5.798
n.skip	200	1	20	10.363	7.748	10.351	7.746	8.512	5.795	8.532	5.808
cbow	200	1	30	10.735	8.143	10.724	8.148	8.592	5.846	8.613	5.859
n.cbow	200	1	30	10.750	8.152	10.733	8.155	8.601	5.853	8.620	5.864
skip	200	1	30	10.338	7.722	10.329	7.727	8.492	5.779	8.512	5.794
n.skip	200	1	30	10.333	7.720	10.319	7.717	8.501	5.789	8.518	5.800
cbow	300	1	30	10.689	8.088	10.675	8.096	8.583	5.837	8.601	5.854
n.cbow	300	1	30	10.687	8.083	10.668	8.084	8.586	5.840	8.603	5.853
ElasticNet											
cbow	100	10	20	10.806	8.203	10.795	8.212	8.702	5.909	8.726	5.926
n.cbow	100	10	20	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
cbow	100	10	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
n.cbow	100	10	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
skip	100	10	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
n.skip	100	10	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
skip	200	1	20	10.949	8.349	10.937	8.359	8.736	5.934	8.760	5.951
n.skip	200	1	20	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
cbow	200	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
n.cbow	200	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
skip	200	1	30	11.026	8.433	11.017	8.445	8.741	5.938	8.766	5.956
n.skip	200	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
cbow	300	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
n.cbow	300	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
GradBoost											
cbow	100	10	20	9.093	6.356	9.066	6.345	8.401	5.699	8.395	5.700
n.cbow	100	10	20	9.043	6.320	9.020	6.313	8.375	5.680	8.367	5.677
cbow	100	10	30	9.294	6.527	9.277	6.529	8.495	5.766	8.491	5.770
n.cbow	100	10	30	9.241	6.487	9.216	6.480	8.481	5.760	8.472	5.758
skip	100	10	30	9.363	6.580	9.342	6.576	8.519	5.785	8.512	5.785
n.skip	100	10	30	9.204	6.458	9.177	6.449	8.456	5.742	8.448	5.742
skip	200	1	20	9.067	6.341	9.043	6.333	8.383	5.682	8.377	5.683
n.skip	200	1	20	8.998	6.290	8.973	6.281	8.350	5.660	8.337	5.658
cbow	200	1	30	9.365	6.583	9.344	6.580	8.520	5.784	8.512	5.785
n.cbow	200	1	30	9.204	6.457	9.176	6.447	8.455	5.743	8.448	5.743
skip	200	1	30	9.048	6.323	9.025	6.316	8.380	5.683	8.371	5.681
n.skip	200	1	30	8.983	6.274	8.965	6.271	8.341	5.656	8.333	5.656
cbow	300	1	30	9.387	6.596	9.367	6.595	8.536	5.799	8.532	5.804
n.cbow	300	1	30	9.172	6.438	9.150	6.430	8.442	5.732	8.430	5.730

[4] N. Arefyev, A. Panchenko, A. Lukanin, O. Lesota, and P. Romanov. Evaluating three corpus-based semantic similarity systems for russian. In *Proceedings of International Conference on Computational Linguistics Dialogue*, 2015.

[5] S. Argamon, S. Dhawle, M. Koppel, and J. W. Pennebaker. Lexical predictors of personality type. In *Proceedings of the 2005 Joint Annual Meeting of the Interface and the Classification Society of North America*, 2005.

[6] Y. Bengio, R. Ducharme, and P. Vincent. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[7] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[8] M. Berggren, J. Karlgren, R. Östling, and M. Parkvall. Inferring the location of authors from words in their texts. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 211–218, Vilnius, Lithuania, May 2015. Linköping University Electronic Press, Sweden.

[9] S. Bergsma and B. Van Durme. Using conceptual class attributes to characterize social media users. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 710–720, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

TABLE VI. A COMPARISON OF THE MEANVEC AND LARGESTCLUSTER ALGORITHMS ON THE EXTENDED DATASET FOR VARIOUS NORMALIZED *word2vec* MODELS

	Word2vec params				Train		Test		Train, nonzero		Test, nonzero	
	type	d	n	v	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
LINEARREGR												
MV	cbow	100	10	20	10.401	7.776	10.397	7.785	8.514	5.792	8.541	5.810
LC	cbow	100	10	20	10.548	7.926	10.542	7.929	8.573	5.834	8.601	5.852
MV	cbow	100	10	30	10.402	7.777	10.396	7.784	8.515	5.791	8.539	5.808
LC	cbow	100	10	30	10.553	7.933	10.551	7.939	8.574	5.832	8.603	5.853
MV	skip	200	1	20	10.738	8.146	10.726	8.151	8.593	5.847	8.613	5.859
LC	skip	200	1	20	10.849	8.251	10.833	8.255	8.629	5.870	8.650	5.883
MV	cbow	200	1	30	10.355	7.737	10.349	7.743	8.497	5.782	8.520	5.798
LC	cbow	200	1	30	10.493	7.878	10.494	7.886	8.554	5.822	8.581	5.839
ELASTICNET												
MV	cbow	100	10	20	10.810	8.208	10.799	8.217	8.694	5.903	8.719	5.921
LC	cbow	100	10	20	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
MV	cbow	100	10	30	10.806	8.203	10.795	8.212	8.702	5.909	8.726	5.926
LC	cbow	100	10	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
MV	skip	200	1	20	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
LC	skip	200	1	20	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
MV	cbow	200	1	30	10.949	8.349	10.937	8.359	8.736	5.934	8.760	5.951
LC	cbow	200	1	30	11.239	8.641	11.229	8.653	8.741	5.938	8.766	5.956
GRADBOOST												
MV	cbow	100	10	20	9.089	6.352	9.065	6.344	8.399	5.697	8.394	5.699
LC	cbow	100	10	20	9.122	6.388	9.100	6.381	8.427	5.720	8.419	5.720
MV	cbow	100	10	30	9.093	6.356	9.066	6.345	8.401	5.699	8.395	5.700
LC	cbow	100	10	30	9.141	6.399	9.120	6.396	8.431	5.720	8.428	5.723
MV	skip	200	1	20	9.363	6.580	9.342	6.576	8.519	5.785	8.512	5.785
LC	skip	200	1	20	9.277	6.520	9.250	6.508	8.493	5.772	8.490	5.770
MV	cbow	200	1	30	9.067	6.341	9.043	6.333	8.383	5.682	8.377	5.683
LC	cbow	200	1	30	9.090	6.361	9.069	6.353	8.406	5.700	8.398	5.702

[10] E. Bloedorn and I. Mani. Using {NLP} for machine learning of user profiles. *Intelligent Data Analysis*, 2(1–4):3 – 18, 1998.

[11] G. Boleda, S. Padó, and J. Utt. Regular polysemy: A distributional model. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 151–160, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[12] S. E. M. E. Bouanani and I. Kassou. Article: Authorship analysis studies: A survey. *International Journal of Computer Applications*, 86(12):22–29, January 2014. Full text available.

[13] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, 1992.

[14] E. Bruni, N. K. Tran, and M. Baroni. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47, 2014.

[15] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[16] Z. Chen, W. Lin, Q. Chen, X. Chen, S. Wei, H. Jiang, and X. Zhu. Revisiting word embedding for contrasting meaning. 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)*, pages 106–115, Beijing, China, 2015. Association for Computational Linguistics.

[17] A. Cimino, F. Dell'Orletta, G. Venturi, and S. Montemagni. Linguistic profiling based on general–purpose features and native language identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 207–215, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[18] R. Cohen and D. Ruths. Classifying political orientation on twitter: It's not easy! In *International AAAI Conference on Weblogs and Social Media*, 2013.

[19] W. Deitrick, Z. Miller, B. Valyou, B. Dickinson, T. Munson, and W. Hu. Gender identification on twitter using the modified balanced winnow. 2012.

[20] N. Djuric, H. Wu, V. Radosavljevic, M. Grbovic, and N. Bhamidipati. Hierarchical neural language models for joint representation of streaming documents and their content. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 248–255, New York, NY, USA, 2015. ACM.

[21] D. Estival, T. Gaustad, S. B. Pham, W. Radford, and B. Hutchinson. Author profiling for english emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING'07)*, pages 263–272, 2007.

[22] L. Flekova and I. Gurevych. Personality profiling of fictional characters using sense-level links between lexical resources. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1805–1816, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[23] P. Forner, R. Navigli, and D. Tufis. Clef 2013 evaluation labs and workshop–working notes papers, 23-26 september, valencia, spain (2013). *URL http://www. clef-initiative. eu/publication/working-notes.*

[24] Y. Goldberg. A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726, 2015.

[25] J. T. Goodman. A bit of progress in language modeling. *Comput. Speech Lang.*, 15(4):403–434, 2001.

[26] R. M. Green and J. W. Sheppard. Comparing frequency-and style-based features for twitter author identification. In *FLAIRS Conference*, 2013.

[27] B. Han, P. Cook, and T. Baldwin. A stacking-based approach to

TABLE VII. A COMPARISON OF THE MEANVEC, LARGESTCLUSTER, AND ALLMEANV ALGORITHMS ON THE BASIC DATASET FOR VARIOUS NORMALIZED *word2vec* MODELS

	Word2vec params				Train		Test	
	type	d	n	v	RMSE	MAE	RMSE	MAE
LINEARREGR								
MV	cbow	100	10	20	8.778	6.066	8.791	6.087
LC	cbow	100	10	20	8.927	6.174	8.946	6.200
AMV	cbow	100	10	20	8.981	6.096	8.991	6.125
MV	cbow	100	10	30	8.798	6.087	8.806	6.095
LC	cbow	100	10	30	8.954	6.201	8.955	6.204
AMV	cbow	100	10	30	8.992	6.108	9.001	6.133
MV	skip	100	10	30	9.051	6.256	9.054	6.268
LC	skip	100	10	30	9.153	6.332	9.146	6.335
AMV	skip	100	10	30	9.186	6.249	9.189	6.270
MV	skip	200	1	20	8.974	6.216	8.983	6.225
LC	skip	200	1	20	9.084	6.292	9.091	6.299
AMV	skip	200	1	20	9.132	6.210	9.137	6.231
MV	cbow	200	1	30	8.734	6.035	8.736	6.052
LC	cbow	200	1	30	8.898	6.158	8.902	6.169
AMV	cbow	200	1	30	8.944	6.071	8.952	6.098
MV	skip	200	1	20	8.976	6.217	8.985	6.226
LC	skip	200	1	20	9.086	6.296	9.094	6.301
AMV	skip	200	1	20	9.133	6.211	9.139	6.232
ELASTICNET								
MV	cbow	100	10	20	9.390	6.498	9.397	6.522
LC	cbow	100	10	20	9.390	6.498	9.397	6.522
AMV	cbow	100	10	20	9.398	6.428	9.398	6.451
MV	cbow	100	10	30	9.390	6.498	9.397	6.522
LC	cbow	100	10	30	9.390	6.498	9.397	6.522
AMV	cbow	100	10	30	9.399	6.428	9.398	6.451
MV	skip	100	10	30	9.390	6.498	9.397	6.522
LC	skip	100	10	30	9.390	6.498	9.397	6.522
AMV	skip	100	10	30	9.399	6.428	9.398	6.451
MV	skip	200	1	20	9.390	6.498	9.397	6.522
LC	skip	200	1	20	9.390	6.498	9.397	6.522
AMV	skip	200	1	20	9.398	6.428	9.398	6.451
MV	cbow	200	1	30	9.390	6.498	9.397	6.522
LC	cbow	200	1	30	9.390	6.498	9.397	6.522
AMV	cbow	200	1	30	9.399	6.428	9.398	6.451
MV	skip	200	1	20	9.390	6.498	9.397	6.522
LC	skip	200	1	20	9.390	6.498	9.397	6.522
AMV	skip	200	1	20	9.399	6.428	9.398	6.451
GRADBOOST								
MV	cbow	100	10	20	8.075	5.398	8.068	5.399
LC	cbow	100	10	20	8.163	5.456	8.172	5.467
AMV	cbow	100	10	20	8.228	5.447	8.275	5.486
MV	cbow	100	10	30	8.086	5.404	8.077	5.403
LC	cbow	100	10	30	8.194	5.479	8.185	5.476
AMV	cbow	100	10	30	8.240	5.456	8.286	5.493
MV	skip	100	10	30	8.277	5.534	8.271	5.537
LC	skip	100	10	30	8.333	5.572	8.336	5.577
AMV	skip	100	10	30	8.362	5.548	8.408	5.583
MV	skip	200	1	20	8.242	5.526	8.231	5.513
LC	skip	200	1	20	8.308	5.562	8.307	5.559
AMV	skip	200	1	20	8.347	5.541	8.392	5.572
MV	cbow	200	1	30	8.032	5.368	8.026	5.366
LC	cbow	200	1	30	8.142	5.447	8.144	5.446
AMV	cbow	200	1	30	8.220	5.443	8.264	5.479
MV	skip	200	1	20	8.240	5.518	8.230	5.510
LC	skip	200	1	20	8.308	5.560	8.302	5.558
AMV	skip	200	1	20	8.347	5.542	8.392	5.573

twitter user geolocation prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[28] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

[29] M. Kågebäck, O. Mogren, N. Tahmasebi, and D. Dubhashi. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, pages 31–39, 2014.

[30] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184 vol.1, 1995.

[31] M. Koppel, J. Schler, S. Argamon, and E. Messeri. Authorship attribution with thousands of candidate. 2006.

[32] V. Lampos, D. PreoÅ£iuc-Pietro, and T. Cohn. A user-centric model of voting intention from social media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 993–1003, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[33] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

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

[35] R. Ling, T. F. Bertel, and P. R. Sundsøy. The socio-demographics of texting: An analysis of traffic data. *New Media & Society*, 14(2):281–298, 2012.

[36] J. Liu and D. Inkpen. Estimating user location in social media with stacked denoising auto-encoders. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 201–210, Denver, Colorado, June 2015. Association for Computational Linguistics.

[37] P. Liu, X. Qiu, and X. Huang. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1284–1290. AAAI Press, 2015.

[38] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun. Topical word embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2418–2424. AAAI Press, 2015.

[39] Q. Luo and W. Xu. Learning word vectors efficiently using shared representations and document representations. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4180–4181. AAAI Press, 2015.

[40] Q. Luo, W. Xu, and J. Guo. A study on the cbow model's overfitting and stability. In *Proceedings of the 5th International Workshop on Web-scale Knowledge Representation Retrieval &#38; Reasoning*, Web-KR '14, pages 9–12, New York, NY, USA, 2014. ACM.

[41] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.

[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[43] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. *INTERSPEECH*, 2:3, 2010.

[44] T. Mikolov, S. Kombrink, L. Burget, J. H. Černockỳ, and S. Khudanpur. Extensions of recurrent neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5528–5531. IEEE, 2011.

[45] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[46] A. Mnih and G. E. Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009.

[47] A. Mnih and K. Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2265–2273. Curran Associates, Inc., 2013.

[48] S. I. Nikolenko and A. Alekseyev. User profiling in text-based recommender systems based on distributed word representations. In *Proc. 5th*

International Conference on Analysis of Images, Social Networks, and Texts, 2016.

[49] J. Oberlander and S. Nowson. Whose thumb is it anyway?: classifying author personality from weblog text. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 627–634. Association for Computational Linguistics, 2006.

[50] W. Paik, S. Yilmazel, E. Brown, M. Poulin, S. Dubon, and C. Amice. Applying natural language processing (nlp) based metadata extraction to automatically acquire user preferences. In *Proceedings of the 1st International Conference on Knowledge Capture*, K-CAP '01, pages 116–122, New York, NY, USA, 2001. ACM.

[51] A. Panchenko, N. Loukachevitch, D. Ustalov, D. Paperno, C. M. Meyer, and N. Konstantinova. Russe: The first workshop on russian semantic similarity. In *Proceedings of the International Conference on Computational Linguistics and Intellectual Technologies (Dialogue)*, pages 89–105, 2015.

[52] T. Pedersen. Screening twitter users for depression and ptsd with lexical decision lists. In *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 46–53, Denver, Colorado, June 5 2015. Association for Computational Linguistics.

[53] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.

[54] D. Preoţiuc-Pietro, V. Lampos, and N. Aletras. 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)*, pages 1754–1764, Beijing, China, July 2015. Association for Computational Linguistics.

[55] M. Qiu, L. Yang, and J. Jiang. Mining user relations from online discussions using sentiment analysis and probabilistic matrix factorization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 401–410, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[56] A. Rahimi, T. Cohn, and T. Baldwin. Twitter user geolocation using a unified text and network prediction model. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 630–636, Beijing, China, July 2015. Association for Computational Linguistics.

[57] A. Rahimi, D. Vu, T. Cohn, and T. Baldwin. Exploiting text and network context for geolocation of social media users. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1362–1367, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[58] F. Rangel and P. Rosso. On the impact of emotions on author profiling. *Information Processing & Management*, 52(1):73 – 92, 2016. Emotion and Sentiment in Social and Expressive Media.

[59] F. Rangel, P. Rosso, M. Moshe Koppel, E. Stamatatos, and G. Inches. Overview of the author profiling task at pan 2013. In *CLEF Conference on Multilingual and Multimodal Information Access Evaluation*, pages 352–365. CELCT, 2013.

[60] F. Rangel, P. Rosso, M. Potthast, B. Stein, and W. Daelemans. Overview of the 3rd author profiling task at pan 2015. In *CLEF*, 2015.

[61] F. Rangel, P. Rosso, M. Potthast, M. Trenkmann, B. Stein, B. Verhoeven, W. Daeleman, et al. Overview of the 2nd author profiling task at pan 2014. In *CEUR Workshop Proceedings*, volume 1180, pages 898–927. CEUR Workshop Proceedings, 2014.

[62] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, and B. Stein. Overview of the 4th author profiling task at pan 2016: cross-genre evaluations. *Working Notes Papers of the CLEF*, 2016.

[63] J. Reisinger and R. J. Mooney. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 109–117, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[64] X. Rong. word2vec parameter learning explained. *CoRR*, abs/1411.2738, 2014.

[65] E. Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009.

[66] E. Stamatatos, W. Daelemans, B. Verhoeven, P. Juola, A. López-López, M. Potthast, and B. Stein. Overview of the author identification task at pan 2015.

[67] D. Tang, B. Qin, and T. Liu. Learning semantic representations of users and products for document level sentiment classification. 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)*, pages 1014–1023, Beijing, China, July 2015. Association for Computational Linguistics.

[68] W. tau Yih, X. He, and C. Meek. Semantic parsing for single-relation question answering. In *Proceedings of ACL*. Association for Computational Linguistics, 2014.

[69] S. Volkova, G. Coppersmith, and B. Van Durme. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 186–196, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[70] Z. Wang, S. LI, F. Kong, and G. Zhou. Collective personal profile summarization with social networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[71] B. Wing and J. Baldridge. Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 336–348, Doha, Qatar, October 2014. Association for Computational Linguistics.

[72] Z. Wu and C. L. Giles. Sense-aware semantic analysis: A multi-prototype word representation model using wikipedia. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2188–2194. AAAI Press, 2015.

[73] W.-t. Yih, G. Zweig, and J. C. Platt. Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1212–1222, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[74] R. Zheng, Y. Qin, Z. Huang, and H. Chen. Authorship analysis in cybercrime investigation. In *Intelligence and Security Informatics*, pages 59–73. Springer, 2003.