# Author Profiling

## Using Deep Learning to Identify Age Groups and Genders

Dario Gjorgjevski[1]
gjorgjevski.dario@students.finki.ukim.mk

[1]Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University in Skopje

November 4, 2017

# Contents

# Motivation

- Huge volume of user-generated content $\Longrightarrow$ appealing to profile users based on it.
- Profiling has multi-faceted advantages:

  Commercial: Analyze the demographics of people that like or dislike certain products.

  Forensic: Find out the linguistic profile of the author of a harassing text message and identify certain characteristics (*language as evidence*).

# Task

### Definition (Author profiling)

Author profiling is the task of identifying certain features of authors of text.

- We will focus on the the *genders* and *age groups* of authors of Twitter posts.
- Genders are either male or female.
- Age groups are 18–24, 25–34, 35–49, 50–64, and 65–xx.

# Dataset

- Data are provided as part of the PAN 2016 event[1].
- There are 277 792 tweets written in English by 436 users, or an average of 637.14 tweets per user.
- 218 users are male and 218 are female.
- The distribution of age groups is given in table 1.
- We additionally used the data provided by the 2013, 2014, and 2017 PAN events for the unsupervised part of our classification pipeline (stay tuned).

[1] http://pan.webis.de/clef16/pan16-web/author-profiling.html

# Age groups

Table: Distribution of age groups

| Age group | Number of users |
|---|---|
| 18–24 | 28 |
| 25–34 | 140 |
| 35–49 | 182 |
| 50–64 | 80 |
| 64–xx | 6 |

One can see that there is class imbalance here. However, we are not going to attempt to rectify this by weighing or resampling.

# Contents

# Parsing

- Data are given in XML files: there is one XML file per author named by the author's hexadecimal ID.
- We used the Beautiful Soup library to parse the files.
- Additionally, there is a file called `truth.txt` which consists of lines of the form

  ```
  <filename>:::<gender>:::<age-group>
  ```

  and gives the ground truth.

# Normalization

1. Hash tags, replies, and external links are replaced by new and unique tokens.
2. Tweets are tokenized into words using the Punkt tokenizer [2] from NLTK, the natural language toolkit.
3. Abbreviations and nonstandard words such as “ur” and “lol” are converted to their standard forms using a dictionary constructed from the ACL 2015 workshop on noisy user-generated content [1].

# Contents

# Distributed representations

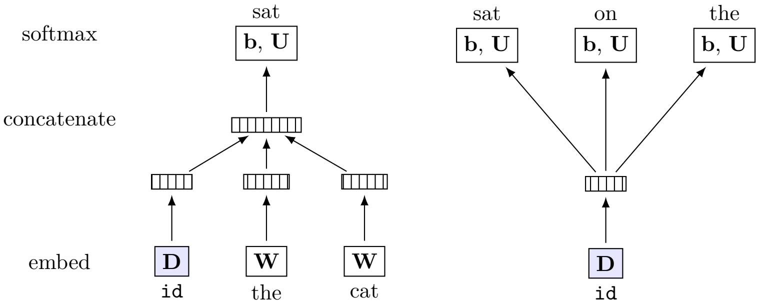
## Motivation

The Doc2Vec model [3] learns distributed representations of both words and entire documents. In our case, document := author.

- Semantically related entities, e.g., Prague, Rome, Berlin, etc., should be close to each other.
- Two entities are considered close if they appear in the same context (Word2Vec [4]).
- The context is a simple window of $k$ words to the left and to the right of a target word.
- The "leftovers" of the context are explained by the *document's ID* itself (Doc2Vec).

# Distributed representations

## Architecture




(a) DM model (b) DBOW model

Figure: Doc2Vec with embeddings **W** and **D**

## Distributed representations

### Learning

- We trained the DM model (figure 1a) using Gensim [5] on 1 047 358 posts (PAN 2013, 2014, and 2017) to learn embeddings of size 256.
- Trained was carried for 20 epochs using a context window of size 8.
- Some examples for closest representations can be seen in listing 1.

# Distributed representations

## Examples

```
>>> model.most_similar(
...    'substantial')

[('higher', 0.72),
 ('large', 0.70),
 ('high', 0.69),
 ('significant', 0.6),
 ('huge', 0.6),
 ('reduced', 0.6),
 ('massive', 0.59),
 ('increased', 0.58),
 ('considerable', 0.57),
 ('enormous', 0.56)]
```

```
>>> model.most_similar(
...    'trump')

[('ivanka', 0.5),
 ('donald', 0.41),
 ('gould', 0.4),
 ('melania', 0.38),
 ('selby', 0.38),
 ('osullivan', 0.35),
 ('finalist', 0.34),
 ('hudson', 0.31),
 ('qingyang', 0.31),
 ('ferguson', 0.31)]
```

Listing 1: Similarities between learned word vectors

# Classification

1. Words are padded to a length of 512, transformed to their distributed representations, and fed to two CNNs:
   1.1 128 filters using a kernel size of 4, rectified linear units (ReLUs) for activation, and max-pooled with a factor of 4.
   1.2 128 filters using a kernel size of 4, ReLUs for activation, and max-pooled with a factor of 16.
2. Author IDs are transformed to their distributed representations and fed to a densely-connected network of 128 ReLUs.
3. The two outputs are concatenated and fed to:
   3.1 Densely-connected layer of 128 ReLUs.
   3.2 Densely-connected layer of 32 ReLUs.
   3.3 Softmax panel for classification.

# Contents

# Results

- Training the architecture for 3 epochs yielded an accuracy of 81.30 % for gender, and 61.18 % for age groups.
- The accuracy was estimated using 1/3 of the data for validation.
- Better results than the ones observed at PAN 2016.

# Future work

- We would like to determine why over-fitting began to occur after the 3rd epoch. It might be worthwhile to apply $L_1$ or $L_2$ regularization, or more aggressive dropout.
- Lexical normalization is not perfect: since a dictionary is used, “ur” will always be translated to “your,” even though it might sometimes stand for “you are.”
- An obvious improvement is to train the distributed representations on more data, and perhaps lemmatize the texts before processing them.

# References I


Bo Han, Paul Cook, and Timothy Baldwin. "Automatically Constructing a Normalisation Dictionary for Microblogs". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 421–432. URL: http://dl.acm.org/citation.cfm?id=2390948.2391000.

Tibor Kiss and Jan Strunk. "Unsupervised Multilingual Sentence Boundary Detection". In: *Comput. Linguist.* 32.4 (Dec. 2006), pp. 485–525. ISSN: 0891-2017. DOI: 10.1162/coli.2006.32.4.485. URL: http://dx.doi.org/10.1162/coli.2006.32.4.485.

# References II


Quoc Le and Tomas Mikolov. “Distributed Representations of Sentences and Documents”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China, June 2014, pp. 1188–1196. URL: http://proceedings.mlr.press/v32/le14.html.

Tomas Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. NIPS’13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119. URL: http://dl.acm.org/citation.cfm?id=2999792.2999959.

# References III

Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50. URL: http://is.muni.cz/publication/884893/en.