INAOE's participation at PAN'13: Author Profiling task

A. Pastor López-Monroy[1], Manuel Montes-y-Gómez[1], Hugo Jair Escalante[1], Luis Villaseñor-Pineda[1], Esaú Villatoro-Tello[2]
Department of Computer Science, Instituto Nacional de Astrofísica, Óptica y Electrónica[1], México
{pastor, mmontesg, hugojair, villasen}@ccc.inaoep.mx
Information Technologies Department, Universidad Autónoma Metropolitana-Cuajimalpa[2], México
evillatoro@correo.cua.uam.mx

1. Introduction

- The Author Profiling (AP) task consists in knowing as much as possible about an unknown author, just by analyzing a given text [2], for example: **age** and **gender**.
- The PAN13 AP task consists in profiling **age** and **gender** in social media data.
- The AP task can be approached as a classification problem.

Differences with other classification tasks are in: i) The used textual features, and ii) The representation.

The standard Bag of Terms (BOT)

Some shortcomings of BOT like representations are:

- High dimensionality.
- High sparseness of the representation.
- They do not preserve any kind of relationship among terms.

Our proposal

- We propose the use of very simple but highly effective meta-attributes.
- These textual features highlight the relationships that terms and documents hold with profiles.
- These attributes are inspired in some ideas from CSA [3] to represent documents in text classification.

2. Document Representation

Document Profile Representation (DPR)

DPR is built in two steps:

1. **Terms representation** in a space of profiles.
2. **Documents representation** in a space of profiles.

	p_1	.	.	.	p_i
d_1	$dp_{11}(p_1, d_1)$	.	.	.	$dp_{i1}(p_i, d_1)$
.	.	.			
.	.		.		
.	.			.	
d_j	$dp_{1j}(p_1, d_j)$				$dp_{ij}(p_i, d_j)$

1) Terms representation

For each term t_j in the vocabulary, we build a term vector $\mathbf{t_j} = \langle tp_{1j}, \ldots, tp_{ij} \rangle$, where tp_{ij} is a value representing the relationship of the term t_j with the profile p_i. For computing tp_{ij} first:

$$wtp_{ij} = \sum_{k:d_k \in P_i} \log_2 \left(1 + \frac{tf_{kj}}{len(d_k)} \right)$$

	p_1	.	.	.	p_i
t_1	$wtp_{11}(p_1, t_1)$	.	.	.	$wtp_{i1}(p_i, t_1)$
.	.	.			
.	.		.		
.	.			.	
t_j	$wtp_{1j}(p_1, t_j)$				$wtp_{ij}(p_i, t_j)$

2. Document Representation

1.1) Normalization

So we get $\mathbf{t_j} = \langle wtp_{1j}, \ldots, wtp_{ij} \rangle$, and finally we normalize each wtp_{ij} as:

$$tp_{ij} = \frac{wtp_{ij}}{\sum_{j=1}^{TERMS} wtp_{ij}} \qquad tp_{ij} = \frac{wtp_{ij}}{\sum_{i=1}^{PROFILES} wtp_{ij}}$$

In this way, for each term in the vocabulary, we get a term vector $\mathbf{t_j} = \langle tp_{1j}, \ldots, tp_{ij} \rangle$.

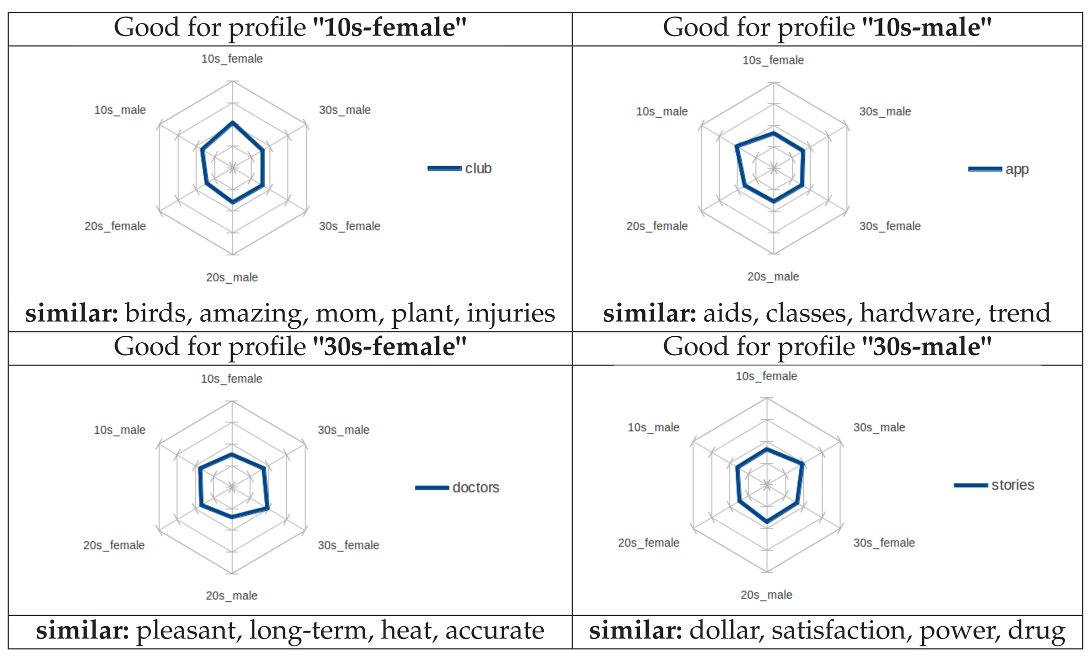
2) Documents representation

Add term vectors of each document. Documents will be represented as $\mathbf{d_k} = \langle dp_{1k}, \ldots, dp_{nk} \rangle$, where dp_{ik} represents the relationship of d_k with p_i.

$$\vec{d_k} = \sum_{t_j \epsilon D_k} \frac{tf_{kj}}{len(d_k)} \times \vec{t_j}$$

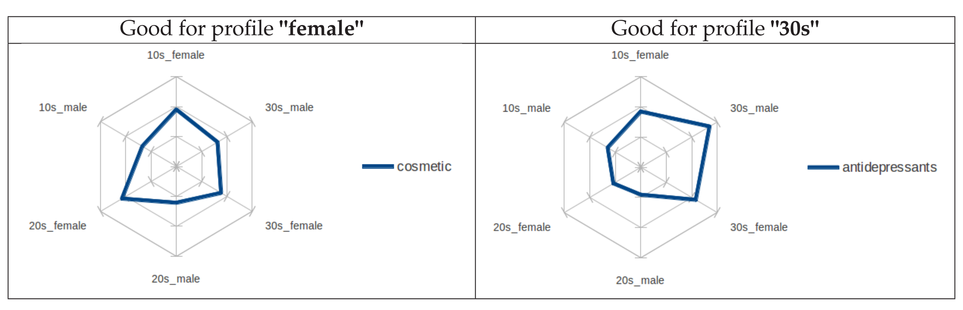
where D_k is the set of terms of document d_k.

Examples of highly descriptive term vectors.



Some term vectors have stronger peaks.

Highly descriptive term vectors for specific profiles.



There are other similar term vectors for specific profiles for example:

- ":)": for detecting young people (e.g. profiles 10s, and 20s).
- "game": for the prediction of males.

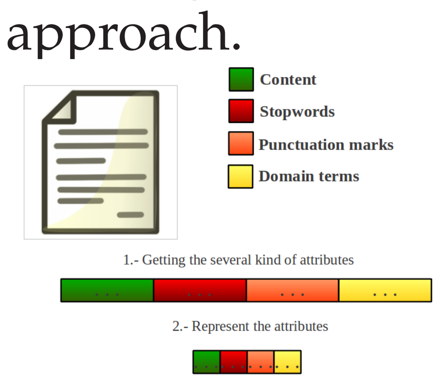
3. Evaluation

Corpus description using our features.

Description for the English corpus according to our textual features							
		Statistics by category					
criteria	**Total**	10s-f	10s-m	20s-f	20s-m	30s-f	30s-m
authors	236600	8600	8600	42900	42900	66800	66800
mean	1058.11	1118.91	1169.02	1005.92	822.75	1172.32	1106.46
std	872.69	918.03	717.56	786.67	918.92	696.84	1021.10
Description for the Spanish corpus according to our textual features							
		Statistics by category					
criteria	**Total**	10s-f	10s-m	20s-f	20s-m	30s-f	30s-m
authors	75900	1250	1250	21300	21300	15400	15400
mean	374.19	234.60	255.36	369	349.044	376.71	434.58
std	704.23	586.42	664.79	586.82	719.41	630.95	884.97

Evaluation

We use the 50K most frequent terms from each information source.
We used a LIBLINEAR classifier [1], and a 10-fold-CV in the training set for preliminary evaluation of our approach.



Final results

Second Order Attributes (SOA) and BOT computed over the 50,000 most frequent terms on the datasets.

Detailed classification accuracy							
	Training data				*Test data*		
	SOA			BOT	SOA		
	Gender	Age	Total	Total	Gender	Age	Total
English	61.3	63.7	**41.9**	36.6	56.90	65.72	**38.13**
Spanish	70.5	72.7	**54.8**	41.9	62.99	65.58	**41.58**

Averaged results for all participants		
AVG		
Gender (st.dv.)	Age (st.dv.)	Total (st.dv.)
53.76 (3.33)	53.51 (12.50)	28.99 (7.42)
55.41 (4.99)	49.04 (14.15)	27.67 (9.35)

Top 5 ranking in the PAN13:

Submission	**Accuracy**			**Runtime**
	Total	Gender	Age	(incl. Spanish)
meina13	0.3894	0.5921	0.6491	383821541
pastor13	0.3813	0.5690	**0.6572**	**2298561**
mechti13	0.3677	0.5816	0.5897	1018000000
santosh13	0.3508	0.5652	0.6408	17511633
yong13	0.3488	0.5671	0.6098	577144695
baseline	0.1650	0.5000	0.3333	–

Submission	**Accuracy**			**Runtime**
	Total	Gender	Age	(incl. English)
santosh13	0.4208	0.6473	0.6430	17511633
pastor13	0.4158	0.6299	**0.6558**	**2298561**
haro13	0.3897	0.6165	0.6219	9559554
flekova13	0.3683	0.6103	0.5966	18476373
ladra13	0.3523	0.6138	0.5727	1729618
baseline	0.1650	0.5000	0.3333	–

4. Conclusions

1. The best method at PAN'13 to predict age profiles in blogs (for both corpus).
2. Our results overcomes the conventional BOT and holds the first position for both languages (overall accuracy).
3. More than 454 times faster than the method in one position below, 166 times faster than the method in first position.
4. This is the first time that AP is addressed using such dense attributes vectors that represent relationships with profiles.

5. References

[1] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[2] M. Koppel, S. Argamon, and A. R. Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412, 2002.

[3] Z. Li, Z. Xiong, Y. Zhang, C. Liu, and K. Li. Fast text categorization using concise semantic analysis. *Pattern Recognition Letters*, 32(3):441–448, 2011.