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

Adrián Pastor Lopez-Monroy, MSc.

M. Montes-y-Gomez, PhD. **H. J. Escalante, PhD.**

L. Villaseñor-Pineda, PhD.

July-2013

Puebla, México

Computer Science Department

Instituto Nacional de Astrofísica, Óptica y Electrónica

Contents

Introduction

- The Author Profiling (AP) task consists in knowing as much as possible about an unknown author, just by analyzing a given text [3].
- Some applications have to do with business intelligence, computer forensics and security.

Introduction

- How much can we conclude about the author of a text simply by analyzing it?
- AP should not be approached exactly in the same way as other text classification (e.g., thematic classification, or authorship attribution).
- Unlike Authorship Attribution (AA), on the problem of AP we does not have a set of potential candidates. Instead of that, the idea is to exploit more general observations (socio linguistic) of groups of people talking or writing [1].
- Initially some works in AP have started to explore the problem of detecting gender, age, native language, and personality.

Introduction

- Different algorithms and strategies have been proposed in order to carry out the AP.
- AP can be approached as a single-label multiclass classification problem, where profiles represent the classes to discriminate.
- From the point of view of text classification, we would have a set of training documents, labeled according to a category (e.g., man and woman).

Introduction

This process usually involves three tasks:

- The extraction of features (style, words, etc..).
- The representation of documents.
- The use of a machine learning method for inducing a classification model.

Note that, in this context, the main difference between Author Profiling and Text Classification lies in the first two steps: i) What characteristics are used?, And ii) How to represent the documents?.

Introduction

In the task of PAN AP 2013, we have two corpus; English and Spanish corpus. The task is to determine the gender (male or female) and age (13-17, 23-27, 33-47) for each document. For the English corpus we have:

- 236,000 instances, each instance is a text file with multiple blogs/posts by the same author.
- In total there are 413.564 blogs/posts and 180 809 187 words.

For Spanish corpus we have:

- 75,900 instances, each instance is a text file with multiple blogs/posts by the same author.
- In total there are 126.453 blogs/posts and 21,824,19 words.

Introduction

Description of the corpus according to the used textual features (words, stopwords, punctuation marks and emoticons).

Description for the English corpus							
		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
min	1	1	1	1	1	1	1
25 %	591	669	692	367	75	701	637
50 %	898	987.5	1176	845	685	1213	959
75 %	1541	1553	1577.25	1535	1434	1567	1557
max	69374	33566	12791	19308	51453	50077	69374

Introduction

Description of the corpus according to the used textual features (words, stopwords, punctuation marks and emoticons).

Description for the Spanish corpus							
		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
min	1	3	1	1	1	1	1
25 %	32	33	21	42	31	30	25
50 %	87	74	53	116	79	80	71
75 %	376	212	174	410	323	403	447.25
max	26163	11629	12257	14507	26163	13869	16529

Representation of documents

One of the most common approaches is the Bag of Terms (BOT)

Some shortcomings of BOT like representations are:

- They produce high dimensionality and high dispersion of the information.
- They do not preserve any kind of relationship of terms.

The method

In this way, we propose the use of an alternative existing method to represent documents, in order to overcome these shortcomings for AP.

- Explore the use of second order attributes for AP task, which will be built using textual features for content, style and domain specific elements.

- As a first approach we propose to bring to the AP task, similar ideas to the Concise Semantic Analysis (CSA) [4] to build document vectors.

Document Profile Representation

- DPR stores textual features of documents in a vector, where the problem of dimensionality is limited by the number of profiles to classify.
- DPR is built in two steps:
  - Building term vectors in a space of profiles.
  - Building document vectors 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	$dp_{1j}(p_1, d_j)$				$dp_{ij}(p_i, d_j)$

Term 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:

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

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

Term normalization

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

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

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

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.

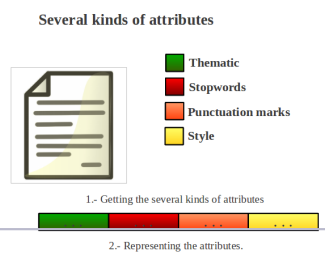
	p_1	.	.	.	p_i
d_1	$dp_{11}(p_1, d_1)$	.	.	.	$dp_{i1}(p_i, d_1)$
.	.	.			
.	.		.		
.	.			.	
d_j	$dp_{1j}(p_1, d_j)$				$dp_{ij}(p_i, d_j)$

Evaluation

The AP task was approached using six *age-gender* profiling classes: *10s-female*, *10s-male*, *20s-female*, *20s-male*, *30s-female*, *30s-male*. To build the representation, the most 50,000 frequent terms were considered (the vocabulary). The considered terms are:

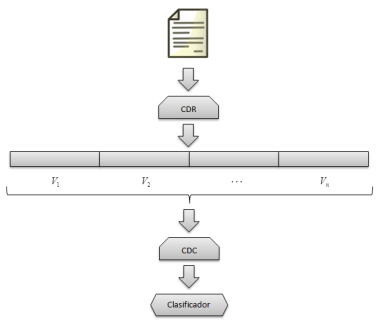
- Words
- Stopwords
- Punctuation marks
- Domain specific vocabulary: e.g., emoticons and hastags.

A LIBLINEAR classifier (similar to linear SVM) was used to perform the prediction [2]. For the experiments, we performed a stratified 10 cross fold validation using the training PAN13 corpus.



Classification of documents

- Represent each instance as a unique vector of second order attributes.
- In this approach, for each document we built the second order attributes using all the vocabularies (e.g., content, stopwords, punctuation marks, etc.)



*Some other tested approaches using SOA were: i) late-fusion vectors, ii) early-fusion vectors, and iii) a very simple instance selection.

Final results

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

Detailed classification accuracy										
	Training data				*Test data*			*Averaged results for all participants*		
	SOA			BOT	SOA			AVG		
	Gender	Age	Total	Total	Gender	Age	Total	Gender (st.dv.)	Age (st.dv.)	Total (st.dv.)
English	61.3	63.7	**41.9**	36.6	56.90	65.72	**38.13**	53.76 (3.33)	53.51 (12.50)	28.99 (7.42)
Spanish	70.5	72.7	**54.8**	41.9	62.99	65.58	**41.58**	55.41 (4.99)	49.04 (14.15)	27.67 (9.35)

Final results for the English corpus

Submission	Accuracy			Adult			Predator			Runtime
	Total	Gender	Age	Gender	Age	Both	Gender	Age	Both	(incl. Spanish)
meina13	0.3894	0.5921	0.6491	6	8	6	72	41	41	383821541
pastor13	0.3813	0.5690	**0.6572**	1	8	0	72	32	32	**2298561**
mechti13	0.3677	0.5816	0.5897	2	6	2	52	29	20	1018000000
santosh13	0.3508	0.5652	0.6408	9	9	9	69	32	29	17511633
yong13	0.3488	0.5671	0.6098	6	1	1	28	30	17	577144695
ladra13	0.3420	0.5608	0.6118	9	9	9	72	33	33	1729618
ayala13	0.3292	0.5522	0.5923	3	2	1	53	34	26	23612726
gillam13	0.3268	0.5410	0.6031	1	4	0	72	30	30	615347
kern13	0.3115	0.5267	0.5690	9	9	9	47	35	25	18285830
haro13	0.3114	0.5456	0.5966	0	8	0	69	44	41	9559554
aditya13	0.2843	0.5000	0.6055	0	0	0	72	40	40	3734665
hidalgo13	0.2840	0.5000	0.5679	0	0	0	72	40	40	3241899
farias13	0.2816	0.5671	0.5061	4	2	1	55	34	26	24558035
jankowska13	0.2814	0.5381	0.4738	1	0	0	72	44	44	16761536
flekova13	0.2785	0.5343	0.5287	4	4	4	61	39	34	18476373
weren13	0.2564	0.5044	0.5099	1	0	0	71	40	39	11684955
ramirez13	0.2471	0.4781	0.5415	9	0	0	12	40	9	64350734
jimenez13	0.2450	0.4998	0.4885	6	2	1	27	31	14	3940310
moreau13	0.2395	0.4941	0.4824	4	4	2	33	39	19	448406705
baseline	0.1650	0.5000	0.3333	–	–	–	–	–	–	–
patra13	0.1574	0.5683	0.2895	5	4	1	55	17	12	22914419
cagnina13	0.0741	0.5040	0.1234	4	7	4	24	9	8	855252000

Final results for the Spanish corpus

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
jimenez13	0.3145	0.5627	0.5429	3940310
kern13	0.3134	0.5706	0.5375	18285830
yong13	0.3120	0.5468	0.5705	577144695
ramirez13	0.2934	0.5116	0.5651	64350734
aditya13	0.2824	0.5000	0.5643	3734665
jankowska13	0.2592	0.5846	0.4276	16761536
meina13	0.2549	0.5287	0.4930	383821541
gillam13	0.2543	0.4784	0.5377	615347
moreau13	0.2539	0.4967	0.5049	448406705
weren13	0.2463	0.5362	0.4615	11684955
cagnina13	0.2339	0.5516	0.4148	855252000
hidalgo13	0.2000	0.5000	0.4000	3241899
farias13	0.1757	0.4982	0.3554	24558035
baseline	0.1650	0.5000	0.3333	–
ayala13	0.1638	0.5526	0.2915	23612726
mechti13	0.0287	0.5455	0.0512	1018000000

Conclusions

1. The proposed approach is the best method at PAN'13 to predict age profiles in blogs (for both corpus).
2. For the six-class AP task at PAN'13, our results overcomes the conventional BOT and holds the first position for both languages (overall accuracy), and second position for each one.
3. For the english corpus, the proposed approach took only 0.22 % (more than 454 times faster) of the time required by the method in one position below, and 0.59 % (more than 166 times faster) of the time required by the method in first position.
4. This is the first time that AP is addressed using attributes that represent relationships with profiles.
5. Through very low computational cost our proposal can build discriminative low dimensional dense vectors for AP

End of the first part.

. . . Questions?

References

Shlomo Argamon, Moshe Koppel, James W Pennebaker, and Jonathan Schler.
Automatically profiling the author of an anonymous text.
Communications of the ACM, 52(2):119–123, 2009.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin.
LIBLINEAR: A library for large linear classification.
Journal of Machine Learning Research, 9:1871–1874, 2008.

Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni.
Automatically categorizing written texts by author gender.
Literary and Linguistic Computing, 17(4):401–412, 2002.

L. Zhixing, X. Zhongyang, Z. Yufang, L. Chunyong, and L. Kuan.
Fast text categorization using concise semantic analysis.
Pattern Recognition Letters, 32:441–448, 2010.