wUGs: Co-Training vs. Simple SVM
Comparing Two Approaches for Cross-Genre Gender Prediction

Lennart Faber, Ian Matroos, Léon Melein, and Wessel Reijngoud

{l.n.faber,i.matroos,l.r.melein,w.reijngoud}@student.rug.nl

Department of Information Science
University of Groningen, The Netherlands

Abstract. The CLIN 29 shared task is concerned with binary gender prediction within and across different genres in Dutch. Our proposed approaches to this problem are a simple model, which uses character n-grams, and a more complex model which consists of two systems in a co-training setup. Both of these approaches beat the baseline scores in all in-genre and cross-genre settings. Our simple model works better in an in-genre setting. The simple model performs only slightly worse in a cross-genre setting than our complex model. This is in line with the findings of Basile et al. (2017). We conclude that our co-training setup does not seem to work as well as expected for cross-genre gender detection. We believe that this might be caused by the fact that the difference between instances is larger between genres than between genders, making initial predictions for the co-training setup inaccurate.

1 Introduction

Gender prediction is a relatively common author profiling task. The theory behind gender prediction is that men and women use slightly different variations of language, and that they write about different subjects. This is also why state-of-the-art models use relatively traditional systems and features, such as n-grams and a Support Vector Classifier (Basile et al., 2017).

Unfortunately, these models do not seem to work well in a cross-genre setting, as the difference in writing between genres is larger than the difference between genders. To tackle this problem, and to determine if a cross-genre gender prediction model is currently even feasible, multiple shared tasks have been organized in the past year. For Italian, there was the EVALITA 2018 Gender x Genre

(GxG) task[1]. For Dutch, the CLIN 29 shared task[2] was organized with a similar setup and datasets.

In this paper, we compare two approaches, a simple approach and a more complex approach. By participating in this shared task, we have tried not only to create a cross-genre gender prediction model, but also to answer our own research question:

Does a simple support vector machine model outperform a co-training model for gender prediction in a cross-genre setting?

Task description Given a (collection of) text(s) from a specific genre, the gender of the author has to be predicted. The task is cast as a binary classification task, with gender represented as F (female) or M (male). Gender prediction will be done in two ways:

- using a model which has been trained on the same genre;
- using a model which has been trained on anything but that genre.

2 Background

A common approach to cross-domain prediction is transfer learning. Transfer learning is the process of training a model on a large dataset for one task and then applying that model on another dataset for a related task. This approach can be useful when there is only a small amount of training data available for the target task, while large corpora exist for the source task. This approach is often used for neural networks and for word embeddings, but can also be applied to other types of more traditional machine learning.

Co-training (also known as co-learning) (Blum and Mitchell, 1998) is similar to transductive transfer learning (Pan et al., 2010). Transductive transfer learning means that the source and target tasks are the same, and the domains are different but related. In our case, the relation between the domains is that all documents are written by a single author, while the differences are the genre and type of content. Just like transductive transfer learning, co-learning can be used for domain adaptation. One advantage of co-training, however, is that the data from the target domain does not have to be annotated.

Co-training uses multiple classifiers with different views of a problem which, similar to transfer learning, train on one set and predict on another dataset. Unlike transfer learning, co-learning does not only attempt to build upon previous knowledge, but also on different views.

The classifiers add the predictions they are relatively sure about from the unlabelled set to the training set of the other classifiers. This makes it possible for the other classifiers to learn about these instances in their way, which might then improve the accuracy on the unlabelled set, which in turn adds more of the unlabelled instances to the training sets.

[1] `https://sites.google.com/view/gxg2018/task`

[2] `https://www.let.rug.nl/clin29/shared_task.php`

For our complex model, we have decided to use lexical normalization as a pre-processing step. Lexical normalization is the task of converting non-standard text (e.g. "Somethign liek dis") to clean text ("Something like this"). This pre-processing task tries to minimize the difference between the different genres in this shared task. By converting text from all genres, we reduce the amount of spelling mistakes, phonetic substitutions, and other errors. This should increase the similarity between the genres, as news most likely has less deviation from standard spelling than the Twitter and YouTube domains. As we wanted our system to work for any genre, in a language agnostic way, we have made sure that every genre was handled in the same way.

Even though this pre-processing step most likely will not affect news articles, as those are more-or-less in standard form already, it could be important for the Twitter and YouTube genres. This is especially the case for Twitter, which has often been used as a subject in text normalization because of its character limit (Han and Baldwin, 2011; Li and Liu, 2012; van der Goot and van Noord, 2017).

Contrary to what one might expect, simple models have so far worked better than complex models for gender prediction. Basile et al. (2017) found that their model performed better when less features were used. This is why we have also created a very simple model, which we will compare to our more complex model, in order to find out if complex models consisting of relatively simple systems also perform less well than simple models themselves.

3 Data

The dataset consists of three genres: news excerpts, Twitter posts and YouTube comments. All genres consist of half female authored, half male authored documents. An overview of the data is given in Table 1.

Genre	Training	Test
News	1,832	1,000
Twitter	20,000	4,914
YouTube	14,744	10,000

Table 1. Number of instances per genre in train and test data.

As we can see, the dataset is relatively small in comparison to other NLP datasets. The length of the instances is also shorter than it would normally be, providing only one tweet per user for Twitter, and one reaction for YouTube users. The news instances are highly variable in length, ranging from only one or two sentences to entire articles. We have also used word embeddings trained on external data. These embeddings have been trained on multiple datasets from different domains, including but not limited to the SONAR 500 corpus, Twente News Corpus (Ordelman et al., 2007) and the 'Geloof der Kamaraden' lectures.

These embeddings are represented as an array containing average embedding vectors per document.

4 Method

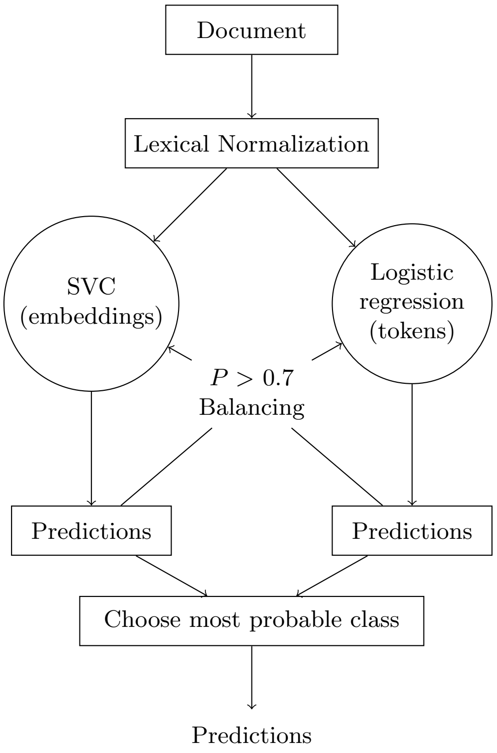

Fig. 1. Our co-training approach

4.1 Pre-processing

For the simple approach, we use no pre-processing. For the more complex approach, lexical normalization is applied to all documents. This increases the likelihood that a certain word would match a known n-gram combination or word embedding. We use the MoNoise system (van der Goot and van Noord, 2017) with the default model for Dutch to perform normalization.

4.2 Approaches

In this section we introduce our simple ($\mathcal{R}_1$) and complex ($\mathcal{R}_2$) approaches to cross-genre prediction. Both rely on *scikit-learn* (Pedregosa et al., 2011) for their system implementations.

$\mathcal{R}_1$ uses 3-6 character n-grams from within word boundaries. We use TF-IDF vectors based on these n-grams to train a Support Vector Machine (SVM) with a linear kernel. This approach aims to not only be robust to slight spelling variations and mistakes, but also tries to generalize well enough.

By using character n-grams instead of the commonly used combination of word and character n-grams, our model will be less likely to accidentally model topic instead of gender. Even though such large n-grams are able to capture smaller words or topics, we did not observe this for the Twitter and YouTube genres during development. For the news genre, we found that some substrings such as 'bier' (beer) occurred multiple times in the list of most significant features, which means that some topic modelling will still have occurred. A side effect of this approach is that some grammatical structures might not be detected by character n-grams alone. One of the phenomena that we cannot detect with our n-grams is the relation between the words.

$\mathcal{R}_2$ uses two different systems and feature sets in a co-training setup. The first system uses binary TF-IDF vectors that represent document tokens, which are fed to a Logistic Regression model. The second system uses external word embeddings as features in a support vector classifier with a linear kernel. A schematic overview is provided in Figure 1.

Each system is trained and predicts classes for an unlabelled dataset. In a cross-genre setting, source data is used as training set, and target data with the labels removed, to provide unlabelled data. In in-genre settings, the training data is evenly split between the training and unlabelled set.

Documents that belong to a certain class with a certainty above a set threshold according to one system, are added to the training data of the other system and vice versa. The training data is then re-balanced, to reduce the chance of overfitting on one class. During development we observed that without balancing, one class would be over-represented in the new training instances. This process continues until no new items are transferred between the training sets of both systems or a maximum number of iterations is reached. If the system finds that it should train on the same sets as during a previous iteration, it will also stop early, to prevent computationally wasteful training loops.

To ensure that instances with a low confidence score will not be added to the training data, we use a threshold (P) to filter these. Empirically, $P = 0.7$ showed the most stable results. For the same reason, we have chosen to limit the maximum number of iterations of the co-learning process. Using five iterations at most gives a good balance between run time and performance on the different genres. Due to (run) time constraints, we could not tune this parameter in a more sophisticated manner.

Target Source	News	Twitter	YouTube
News		0.507 / 0.508	0.503 / 0.501
Twitter	**0.550 / 0.547**		**0.523 / 0.524**
YouTube	0.536 / 0.532	**0.541 / 0.539**	

Table 2. Scores on the development set in cross-genre settings, with normalized / non-normalized data.

5 Results

5.1 Model selection

For the cross-genre scores, we based our choice on the performance of different setups on the development data. To validate our models, we split our training data evenly into a training and development set. Half of this data was used as unlabelled data for our co-training setup, while the other half was used as test data. Training was done with the full training set of another genre. Based on the scores in Table 2, we decided to use normalization for our co-training approach as it performed better on two of the three genres. We selected the best performing model for each target genre. These models are shown in bold in Table 2.

For our simple approach we did not consider using lexical normalization as we wanted to compare our co-training approach to a simple model.

The results of both systems are presented in Table 3. $\mathcal{R}_1$ performs better in-genre, outperforming $\mathcal{R}_2$ in every genre. $\mathcal{R}_2$ performs best on average in the cross-genre setting, although the difference with $\mathcal{R}_1$ is small.

Compared to other submissions of the CLIN shared task, the results of $\mathcal{R}_1$ are interesting. On the in-genre task, the model ranks second. This confirms the conclusion of Basile et al. (2017) that simple traditional models still perform very well on this task.

Genre	IN		CROSS	
	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_1$	$\mathcal{R}_2$
News	**0.6890**	0.5830	0.5260	**0.5530**
Twitter	**0.6367**	0.6241	**0.5406**	0.5376
YouTube	**0.6156**	0.5849	**0.5360**	0.5212
Average	**0.6471**	0.5973	0.5342	**0.5373**

Table 3. Accuracy scores on the test sets. in-genre and cross-genre. The cross-genre results were obtained by submitting the models that performed best on the training data.

6 Discussion

The results of the co-training setup were somewhat disappointing. We believe that the cause for this lies with the initial predictions the system uses. The differences between instances appear to be larger between the two genres than the different genders. As a result, the system is fed with wrong information and is also not able to overcome the lack of new information.

Some aspects of the co-training setup that could have been improved further. The parameters regarding transfer of training instances and the maximum number of iterations were set based on balancing (run) time with the results on the different genres. Fine-tuning these parameters could lead to better results.

The simple model is also limited by its feature set. As we use word boundary n-grams, almost similarly spelled words result in almost similar n-grams, which results in the loss of grammatical information. Rangel and Rosso (2016) suggest that certain morphosyntactic information is reasonably indicative of gender. Extending the n-grams beyond word boundaries could help to also capture this information.

7 Future work

As our systems did not reach similar accuracy between the in-genre and cross-genre settings, we have not managed to create a domain-agnostic model. We believe that more work towards this task should be performed. One of the most important steps would be assessing how well human annotators can perform this task. This could shed some light on whether better results can reasonably be expected or not.

If human evaluators do manage to score significantly higher than current systems, we suggest focusing on simple approaches. These approaches seem to work just as well as a co-training approach and are often easier and faster to train.

Acknowledgments

We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster. We also want to thank Rob van der Goot for both continuous feedback and his support running the MoNoise model for Dutch. The word embeddings that we used in our complex model were trained by Gertjan van Noord.

Bibliography

Angelo Basile, Gareth Dwyer, Masha Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim. 2017. N-gram: New groningen author-profiling model. Conference and Labs of the Evaluation Forum (CLEF 2017) : Information Access Evaluation meets Multilinguality, Multimodality, and Visualization ; Conference date: 11-09-2017 Through 14-09-2017.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Rob van der Goot and Gertjan van Noord. 2017. Monoise: Modeling noise using a modular normalization system. *Computational Linguistics in the Netherlands Journal*, 7:129–144.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 368–378. Association for Computational Linguistics.

Chen Li and Yang Liu. 2012. Normalization of text messages using character-and phone-based machine translation approaches. In *Thirteenth Annual Conference of the International Speech Communication Association*.

Roeland J.F. Ordelman, Franciska M.G. de Jong, Adrianus J. van Hessen, and G.H.W. Hondorp. 2007. Twnc: a multifaceted dutch news corpus. *ELRA Newsletter*, 12(3-4).

Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincest Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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