A Semi-Supervised Approach for Gender Identification

Juan Soler-Company[1], Leo Wanner[1,2]
[1]NLP Group, Department of Information and Communication Technologies, Pompeu Fabra University,
[2]Catalan Institute for Research and Advanced Studies (ICREA)
[1]C/ Roc Boronat, 138, 08018 Barcelona, Spain
juan.soler@upf.edu, leo.wanner@upf.edu

Abstract

In most of the research studies on Author Profiling, large quantities of correctly labeled data are used to train the models. However, this does not reflect the reality in forensic scenarios: in practical linguistic forensic investigations, the resources that are available to profile the author of a text are usually scarce. To pay tribute to this fact, we implemented a Semi-Supervised Learning variant of the k nearest neighbors algorithm that uses small sets of labeled data and a larger amount of unlabeled data to classify the authors of texts by gender (man vs woman). We describe the enriched KNN algorithm and show that the use of unlabeled instances improves the accuracy of our gender identification model. We also present a feature set that facilitates the use of a very small number of instances, reaching accuracies higher than 70% with only 113 instances to train the model. It is also shown that the algorithm performs equally well using publicly available data.

Keywords: author profiling, gender identification, semi supervised learning, text classification, machine learning

1. Introduction

Author profiling and author gender identification are increasingly popular research areas in Computational Linguistics. The range of potential applications spans from forensic investigations to online marketing studies. In the field of forensic investigations, author profiling can be used, for instance, for pedophile detection in chat rooms, terrorist activity detection monitoring forum/chat/email data, plagiarism detection, etc. In the field of online marketing studies, author profiling can be used to analyze customer feedback data and profile the most active users of the services of online companies.

The field of author gender identification presupposes that men and women think, talk and as a result, write differently, such that gender-distinctive linguistic patterns can be extracted to differentiate between genders. In analogy, the broader field of Author Profiling is based on the assumption is that authors with similar demographic, social, cultural and gender characteristics express themselves following common patterns that can be analyzed to classify their writings with respect to these characteristics.

The majority of works in author profiling use Supervised Learning which requires a sufficiently large corpus of clean, correctly annotated training data. However, in many author profiling tasks, such data is not available. Consider, for instance, forensic applications, where only a limited number of writings of the same author can be counted on, or literature studies, where the amount of written material might be sufficient, but not annotated. In this context, semi-supervised learning (or even unsupervised learning) suggests itself as an alternative. The goal of semi-supervised learning is to use unlabeled data and a small sample of labeled data to learn.

In what follows, we present a semi-supervised variant of the k nearest neighbors (kNN) algorithm for author gender identification. The algorithm uses unlabeled data to enrich the training set. The algorithm assigns the unlabeled data instances estimated gender scores, such that each of these instances can be exploited as additional training data for the corresponding gender class. We show that this strategy helps to boost the performance of the model. The proposed algorithm can be useful in realistic cases in which the correctly labeled data is scarce and the unlabeled data is easy to obtain.

In the next section, we present a short overview of the related work. In Section 3, we outline our semi-supervised variant of kNN. Section 4 describes the set up of the experiments that we carried out to assess the quality of the proposed algorithm, and Section 5 discusses the results we obtain within these experiments. Section 6, finally, draws some conclusions and outlines the lines of our future work in the area of author profiling.

2. Related Work

In the vast majority of the existing works, author profiling and author gender identification are defined as supervised machine learning problems. Different kinds of data as input have been used; see, among others, (Argamon et al., 2009; Argamon and Shimoni, 2003; Schler et al., 2006; Zhang and Zhang, 2010), where gender, age, native language and personality detection are performed using blog posts and the international corpus of English learners. In (Burger et al., 2011), gender identification is performed for Tweets, and in (Estival et al., 2007) and (Cheng et al., 2009) author profiling is performed on email data. Chat messages have also been worked on. For instance, Kucukyilmaz et al. (2006) and Kose et al. (2008) attempt to extract the gender of the users of chat blogs. Both predict the gender of the members of conversations (in Turkish) in different chat services. Some of the specific features that they use for this purpose are the occurrence of the "smiley" symbol, abbreviations, slang words, and different function words. Gupta et al. (2012) apply this kind of approach to identify pedophiles in chat services.

In (Groom and Pennebaker, 2005), a study on the effects of sexual orientation in the writing of authors is presented. Very relevant for the progress of the state-of-the-art in the field is the yearly shared task on author profiling and similar applications such as plagiarism detection and author obfuscation is held; cf. (Stamatatos et al., 2015a; Stamatatos et al., 2015b; Stamatatos et al., 2015c; Hagen et al., 2015) for more information on the shared tasks.

However, hardly any work has been done on approaching author profiling as a semi-supervised machine learning problem—although semi-supervised learning has been widely used in a number of areas of Computational Linguistics; see e.g, (Zhu, 2005) for a literature survey, (Wong et al., 2008) for the use of semi-supervised learning in text summarization, (Niu et al., 2005) for its application to word sense disambiguation, (Koo et al., 2008) for the use in parsing, (Zhang and Ostendorf, 2012) for classification of movie reviews and newsgroup articles, etc.

3. Enriched KNN algorithm

Our semi-supervised learning algorithm for gender identification is a modified version of the classic *k nearest neighbors* (kNN) classifier. Given a test instance, this algorithm, identifies the k instances that are closest (in accordance with a vector distance metric such as cosine or Euclidean distance) to the test instance. The test instance is labeled with the most common label among its k neighbors.

Our algorithm works in two phases: the *training set enrichment phrase*, and the *classification phase*. In both phases, the values of the features used for the classification task are normalized between 0 and 1. Prior to the classification of an instance, both the test instance and the training set instances are normalized by dividing each feature value by its maximum feature value among all involved instances (i.e., training set instances and the instance that is being classified). Using this strategy makes the computed distances meaningful in the vector space that is being used.

For the normalization, the Euclidean distance between two instances is divided by the number of features. The idea behind this procedure is that since all the features are scaled between 0 and 1, the maximum value that the Euclidean distance can achieve is the number of dimensions of the vectors. The division of this value by the number of features scales the distance between the same boundaries and as a result, the scores are also scaled in the same way.

3.1. Training set enrichment phase

The goal of the algorithm of the enrichment phase (see Algorithm 1) is to expand the initial dataset by giving the unlabeled instances a score for each possible label, and ensuring at the same time that these scores are lower than the ones that the labeled instances have (the labeled instance score will be the upper bound of the unlabeled scores).

Given an unlabeled instance, we obtain the k nearest neighbors, which will be the k labeled instances that have the least Euclidean distance between them, and the given test instance (see Table 5 for the performance of the model using different distance metrics). The unlabeled instances that have been assigned a score are not considered as possible "neighbors", since this would likely lead to much more noise in the enriched dataset due to the fact that the decisions would be made depending on unreliable data.

Algorithm 1 Enrichment Phase

for u in unlabeled_set **do**
  $kneighbors$ = getNearestNeighbors(u, train_set, K)
  $scores$ = dict()
  for n in $kneighbors$ **do**
    $scores$[n.label] = $scores$[n.label] + (n.score[n.label] − n.distance)/K
  end for
  u.setScores($scores$)
  train_set.add(u)
end for

For each unlabeled instance, we increase the score of the label that corresponds to the neighbor by the difference between the neighbor's score (which will be 1.0 because it is a labeled instance) and the Euclidean distance between them divided by K. Depending on the labels of the neighborhood around the unlabeled instance, a probability for each of the possible labels is computed. Using the difference between score and distance to compute the unlabeled instance's score is a way of giving higher scores to instances that are closer (thus, more similar) to the unlabeled one.

After setting the computed scores and adding the new instance to the training set, the labeled instances will have better scores than the unlabeled ones. By default, every instance that is manually labeled will have a score of 1.0 for their correct label and 0.0 for the incorrect one. The scores represent the probability that an instance has a particular label. This is a way to make the unlabeled instances useful while prioritizing the correctly labeled ones. This process of assigning probability-based labels can help classification processes in which the manually labeled data is scarce.

3.2. Classification Phase

The *classification phrase* of the algorithm is outlined in Algorithm 2.

Algorithm 2 Classification Phase

for t in test_set **do**
  $kneighbors$ = getNearestNeighbors(t, train_set, K)
  $scores$ = dict()
  for n in $kneighbors$ **do**
    for $label$ in n.score.keys() **do**
      $scores$[$label$] = $scores$[$label$] + n.score[$label$]
    end for
  end for
  t.label = getMaxLabel($scores$)
end for

To classify the test instances, first of all, the k nearest neighbors are retrieved in the same way as it was done during the *enrichment phase*. Then, for each neighbor, the probabilities for each possible class are added. The class with a better accumulated score provides the label for the test instance. The impact of a manually labeled instance in the neighborhood of a test instance will always be higher

than the impact of the instances that were added in the first phase.

4. Experimental Setup

In this section, we present the dataset on which the experiments were performed and the feature set that was used.

4.1. Dataset

The texts that compose the dataset that we use in our experiments are journalistic opinion columns in which the writer expresses his/her opinion about different topics such as economics, current news or politics. These columns were crawled from newspaper blogs. In total, 1136 posts were crawled from different sources. Table 1 shows the sources from which the opinion columns were retrieved.

Source Name
Dallas News
NYDaily News
Canberra Times
The Telegraph
The Guardian
The Independent

Table 1: Data source list

Half of the texts in the retrieved dataset were written by men and the other half by women (only texts with only one author were considered). These texts were automatically crawled, cleaned from boilerplates and manually tagged by the gender of their author.

In the performed experiments, 113 texts (10%) were used as the initial training set (with known annotations), 113 texts (10%) as test set and the rest of the dataset as unannotated data.

4.2. Feature Set

State-of-the-art author profiling/gender identification proposals use mostly surface-oriented features: function words, most frequent words, triples and/or pairs of frequently co-occurring words, part of speech (POS) n-grams, punctuation marks, etc. Syntactic features are less often used; cf., e.g., (Cheng et al., 2009). However, from linguistics and philology we know that syntactic idiosyncrasies are also distinctive features of the writing style of individuals and groups of individuals who share demographic, social, cultural or gender characteristics; see, e.g., (Crystal and Davy, 1969; Biber, 1989; Strunk and White, 1999; Tufte, 2006). Therefore, syntactic features play in our setup an important role. Similar feature sets were used in (Soler and Wanner, 2014; Soler and Wanner, 2015), where the authors performed gender identification for blog posts as well as gender and language identification in a multilingual scenario.

The features have been extracted using the programming language Python and its Natural Language Toolkit[1]. For the extraction of the syntactic dependency features (see below), the dependency parser from the MATE-tools has been used (Bohnet, 2010).

Each post in the dataset is represented as a multidimensional vector in which each dimension captures the value of a specific feature.

In total, five types of features are used:

Character-based Features that are composed of the ratios between upper cased characters, periods, commas, parenthesis, exclamations, colons, number digits, semicolons, hyphens and quotation marks on the one side and the total number of characters in a post on the other side.

Word-based Features that are composed of the mean numbers of characters per word, vocabulary richness, acronyms, stop words, first person pronouns (both first person singular and plural), ratio between words composed by 2 or 3 characters and the total number of words in a post, standard deviation of word length and the difference between the longest and shortest word.

Sentence-based Features that are composed of the mean numbers of words per sentence, standard deviation of words per sentence and the difference between the maximum and minimum number of words per sentence.

Dictionary-based Features that are composed of the ratios of discourse markers, interjections, abbreviations, curse words, and polar words (differentiating between positive/negative words as well as words that inspire other sentiments such as indifference, sadness, fear, etc. (Staiano and Guerini, 2014)) listed in dictionaries – again with respect to the total number of words in a post.

Syntactic Features that are composed of dependency syntactic features. In particular, the mean depth and width of the dependency trees in a post and the frequencies of individual relations in the dependency trees of the sentences in a post. The depth of the trees is defined as the longest path between the root and one of the leaves. The width is the maximum number of siblings at any of the depths of the tree. The depth and width of dependency trees can be interpreted as a measure of the complexity of the structure of the corresponding sentences. The mean number of different dependency relations used per sentence is also measured. This group of features is the largest one of the presented (it accounts for more than 50% of the total number of features).

5. Results and Discussion

To evaluate the effectiveness of our algorithm and the chosen feature set, we designed two series of experiments. In the first series of experiments, we tested its performance on our genuine task of author gender recognition. In the second series of experiments, we applied it to different datasets.

5.1. Author gender recognition

Two main experiments were carried out in the context of author gender recognition. As already mentioned above, in both, 10% of the dataset were used as training set, another 10% as test set and the rest as unlabeled instances.

To test the behavior of the feature set, we executed first only the *classification phase* (as outlined in Algorithm 2), using

[1] http://nltk.org

10% of the dataset for training and another 10% for testing. Both sets contained the same number of instances per class. To evaluate the accuracy, the classification was executed 1000 times, changing randomly the training and test set in each execution. Table 2 displays the accuracy of the classifier for different ks, comparing it to three baselines that follow the "bag of words" approach and that consist in using the frequencies of the 300, 400 and 500 most common words in the training set for classification.

K	Accuracy	BoW300	BoW400	BoW500
12	**74.19%**	66.81%	66.61%	64.39%
22	**72.80%**	66.88%	65.67%	62.74%
27	**71.06%**	64.77%	63.20%	59.49%
34	**69.51%**	65.89%	63.49%	59.56%
45	**69.47%**	62.77%	59.61%	56.10%
67	**65.39%**	59.65%	56.34%	54.32%

Table 2: Accuracy of the classification phase

We can observe that even if our classifier has only 113 instances for training and the same amount of instances for testing, the accuracies are quite good. To reach more than 70% of accuracy in these conditions is a good indicator that the chosen feature set is effective in distinguishing the gender of the authors.

To have a better understanding of the performance of our feature set and to see which features distinguish better between genders, we computed the information gain coefficients of the features. Table 3 shows the 20 most distinctive features (the features that are upper cased are frequencies of syntactic dependencies).

Feature Name
Vocab. Richness
Interjections
HYPH
TMP
2-character words
Upper cased chars
Word STD
Quotations
Negative Words
Dot frequency
Chars. per Word
First Person Singular Pronouns
Semicolons
Acronyms
Tree Width
VC
NMOD
LOC
Abbreviations
HMOD

Table 3: 20 most distinctive features

Some conclusions can be drawn from the list with the most distinctive features. First of all, it can be observed that syntactic features are very relevant: several dependencies are very distinctive. The width of the syntactic trees is also relevant. This measure can be seen as an indicator that the complexity of the discourse between genders differs. At the word level, we can say that the vocabulary richness and the number of characters per word are also relevant for gender classification.

It is also interesting to note that the percentage of negative words is equally relevant. This can be explained by the hypothesis that, in general, men tend to be less emotionally involved in the stories they write than women. The differences in the usage of first person pronouns is also noticeable. To explain this difference it could be hypothesized that men and women diverge in the degree of the tendency to write about themselves (rather than about people around them).

The second experiment measured the accuracy improvement that is obtained by executing both phases. First, the *classification* is executed, as it was done in the previous experiment. After that, the *enrichment phase* is executed and finally, the *classification* is run again with the enriched dataset (this process was also carried out 1000 times, randomizing trainining and test set each time). Table 4 shows the achieved improvements in accuracy.

K	Accuracy, initial	Accuracy, enriched
12	74.19%	76.82%
22	72.80%	74.32%
27	71.06%	73.28%
34	69.51%	72.69%
45	69.47%	71.88%
67	65.39%	69.89%
80	60.01%	68.04%
100	53.99%	66.96%

Table 4: Accuracy of the combined classification and enrichment phases

We can observe that our classification algorithm achieves good accuracy already with a small sample of instances for training. We believe that this is due to the composition of the feature set we use. However, adding more instances in a semi-supervised fashion lets the classification further improve. More precisely, by adding 863 unlabeled instances, our algorithm improves for every k. Note that in the case of a considerably higher number of unlabeled data and the same number of labeled data, an instance selection process would be required to avoid introducing noise into the training set.

A simple instance selection could consist in the analysis of the standard deviation of the scores for each class of the unlabeled instances; the instances with a higher standard deviation in their score than a threshold would be added, the others would be discarded. For instance, if we have an instance with p(male)=0.55, p(female)=0.45 and an instance with p(male)=0.2, p(female)=0.8, the second instance would be considered as clearly more useful. In small datasets it might not be that problematic, but in the case of thousands of instances, the instances with probabilities as in the first example would introduce a noise that could make

the accuracy of the enriched classification decrease.
It can be also observed that the value of k is very significant for the classification. Higher values of k give the algorithm more information, but they can also make the classification more susceptible to noise and overfitting (also increasing the computational cost). The higher the number of analyzed neighbors, the higher the chance of finding that one or more of the neighbors are actually outliers or unlabeled instances labeled with a low reliability score. Lower values of k make the classification perform better. However, in bigger datasets, these values should be scaled accordingly, otherwise we would be analyzing local neighborhoods that might be too small to be representative.
To analyze the behavior of the described algorithm in its application to gender identification in more detail, we carried out an additional experiment in that we tested the algorithm using other metric distances in order to see how different ways to compute vector distance affect the accuracy and improvement of the model. In Table 5, we see the performance of the model for different distance metrics and different values of k. The displayed accuracies were computed as before, using 1000 random samplings of trainining and test sets, where each one of these sets constituted 10% of the dataset and the rest was used as unlabeled data.

Metric	K	Acc., initial	Acc., enriched
Euclidean	12	74.19%	76.82%
	34	69.51%	72.69%
	80	60.01%	68.04%
Cosine	12	74.29%	75.96%
	34	70.09%	71.85%
	80	60.91%	68.38%
Manhattan	12	77.32%	79.21%
	34	71.88%	75.36%
	80	61.16%	70.23%
Chebyshev	12	60.70%	66.41%
	34	58.18%	64.5%
	80	52.90%	61.44%

Table 5: Performance using different distance metrics

According to the figures in Table 5, for each combination of k and distance metric, the accuracy increases after the enriching phase. This means that the algorithm is working as intended and that the enrichment process is effective. It can also be seen that the Manhattan distance achieves higher values of accuracy.

5.2. Application to other domains

In order to assess whether our algorithm is equally performative when used for other applications, we applied it to two datasets publicly available in (Lichman, 2013): the *Banknote Authentication Data Set*[2] ("bank" from now on) and the *Image Segmentation Data Set*[3] ("image" from now on). These datasets are provided with lists of instances and their extracted feature values. The bank dataset consists of data that were extracted from images that were taken for the evaluation of an authentication procedure for banknotes. It is composed by 1372 instances, which have 4 features and 2 possible classes. The image dataset contains data described by high-level numeric-valued attributes. This dataset has 2315 instances, with 19 features and 7 different classes.
To evaluate the performance of our model with these datasets as input, the same method as before was used: 1000 random samplings with 10% of the data were used as training set, the same amount was as test set, and, finally, the rest was used as unlabeled data. The chosen distance metric was the Euclidean distance. Table 6 displays the performance of the algorithm for both datasets for different values of k.

Dataset	K	Accuracy, initial	Accuracy, enriched
bank	12	91.05%	94.07%
	34	83.68%	87.91%
	80	74.15%	83.04%
image	12	79.06%	83.49%
	34	70.32%	75.65%
	80	60.22%	72.22%

Table 6: Performance on publicly available data

We can see that the algorithm performs well in these datasets as well. The most important message we can derive from the obtained figures is that for every combination of dataset and k, the accuracy after the enrichment phase is higher than before the enrichment phase. This means that using unlabeled data to guide the classification is effective and useful not only in the author profiling case that was presented before.

6. Conclusions and Future Work

We presented a semi-supervised approach to gender identification in which unlabeled data is enriched by being probabilistically labeled depending on their neighborhood. Using small samples of labeled data and the enriched unlabeled instances, the performance of the process is improved.
We also showed that this approach equally works for other applications. Furthermore, we discussed a set of features that achieved very competitive accuracy on scarce training data and analyzed which features of the presented feature set are most distinctive.
In the future, we will apply the presented approach to author profiling in general (not only to gender identification). Furthermore, we are working on sexual orientation detection and the implementation of clustering (i.e., totally unsupervised) models for author profiling. We also plan to participate in the PAN Author Profiling task (Stamatatos et al., 2015b), to evaluate our approach and our feature set against other participants in the same setup.

7. Acknowledgements

The presentation of this work was partially supported by the ICT PhD program of Universitat Pompeu Fabra through a travel grant.

References

Argamon, S. and Shimoni, A. R. (2003). Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17:401–412.

[2] http://archive.ics.uci.edu/ml/datasets/banknote+authentication
[3] http://archive.ics.uci.edu/ml/datasets/Image+Segmentation

Argamon, S., Koppel, M., Pennebaker, J. W., and Schler, J. (2009). Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2):119.

Biber, D. (1989). A typology of english texts. *Linguistics*, 27:3–43.

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, COLING '10, pages 89–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. (2011). Discriminating gender on twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '11, pages 1301–1309, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cheng, N. C. N., Chen, X. C. X., Chandramouli, R., and Subbalakshmi, K. P. (2009). Gender identification from E-mails. *2009 IEEE Symposium on Computational Intelligence and Data Mining*.

Crystal, D. and Davy, D. (1969). *Investigating English Style*. Longman Group Ltd., London.

Estival, D., Gaustad, T., Pham, S. B., Radford, W., and Hutchinson, B. (2007). Author Profiling for English Emails. In *10th Conference of the Pacific Association for Computational Linguistics (PACLING 2007)*, pages 262–272.

Groom, C. J. and Pennebaker, J. W. (2005). The language of love: Sex, sexual orientation, and language use in online personal advertisements. *Sex Roles*, 52(7-8):447–461.

Gupta, A., Kumaraguru, P., and Sureka, A. (2012). Characterizing pedophile conversations on the internet using online grooming. *arXiv preprint arXiv:1208.4324*.

Hagen, M., Potthast, M., and Stein, B. (2015). Source Retrieval for Plagiarism Detection from Large Web Corpora: Recent Approaches. In *Working Notes Papers of the CLEF 2015 Evaluation Labs*, CEUR Workshop Proceedings. CLEF and CEUR-WS.org, September.

Koo, T., Carreras Pérez, X., Collins, M., et al. (2008). Simple semi-supervised dependency parsing.

Kose, C., Ozyurt, O., and Ikibas, C. (2008). A Comparison of Textual Data Mining Methods for Sex 2 A Summary of Turkish Chat Language. pages 638–643.

Kucukyilmaz, T., Cambazoglu, B. B., Aykanat, C., and Can, F. (2006). Chat mining for gender prediction. In Tatyana M. Yakhno et al., editors, *ADVIS*, volume 4243 of *Lecture Notes in Computer Science*, pages 274–283. Springer.

Lichman, M. (2013). UCI machine learning repository.

Niu, Z.-Y., Ji, D.-H., and Tan, C. L. (2005). Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 395–402. Association for Computational Linguistics.

Schler, J., Koppel, M., Argamon, S., and Pennebaker, J. W. (2006). Effects of age and gender on blogging. In *Proceedings of the AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 199–205. AAAI.

Soler, J. and Wanner, L. (2014). How to use less features and reach better performance in author gender identification. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Soler, J. and Wanner, L. (2015). Multiple language gender identification for blog posts. In *Proceedings of the 37th Annual Cognitive Science Society Meeting (COGSCI'15)*, Pasadena, California, EEUU.

Staiano, J. and Guerini, M. (2014). Depechemood: a lexicon for emotion analysis from crowd-annotated news. *CoRR*, abs/1405.1605.

Stamatatos, E., amd Ben Verhoeven, W. D., Juola, P., López-López, A., Potthast, M., and Stein, B. (2015a). Overview of the Author Identification Task at PAN 2015. In Linda Cappellato, et al., editors, *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France*. CEUR-WS.org, September.

Stamatatos, E., Potthast, M., Rangel, F., Rosso, P., and Stein, B. (2015b). Overview of the pan/clef 2015 evaluation lab. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 518–538. Springer.

Stamatatos, E., Potthast, M., Rangel, F., Rosso, P., and Stein, B. (2015c). Overview of the PAN/CLEF 2015 Evaluation Lab. In Josiane Mothe, et al., editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 6th International Conference of the CLEF Initiative (CLEF 15)*, pages 518–538, Berlin Heidelberg New York, September. Springer.

Strunk, W. J. and White, E. (1999). *The Elements of Style, Fourth Edition*. Macmillan, Toronto.

Tufte, V. (2006). *Artful Sentences: Syntax as Style*. Graphics Press, Cheshire.

Wong, K.-F., Wu, M., and Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 985–992. Association for Computational Linguistics.

Zhang, B. and Ostendorf, M. (2012). Semi-Supervised Learning for Text Classification using Feature Affinity Regularization. In *Proceedings of Symposium on Machine Learning in Speech and Language Processing (MLSLP)*.

Zhang, C. and Zhang, P. (2010). Predicting gender from blog posts. *Technical Report. University of Massachusetts Amherst, USA*.

Zhu, X. (2005). Semi-supervised learning literature survey. *world*, 10:10.