THE PAST, PRESENT AND FUTURE OF TEXT CLASSIFICATION

NIKLAS ZECHNER
DEPARTMENT OF COMPUTING SCIENCE
UMEÅ UNIVERSITY
zechner@cs.umu.se

Abstract. Despite over a century of research, the study of text classification is still chaotic. In this article, we give an overview of some of the techniques that have been used, for author identification and for other aspects of classification. The multitude of parameters which vary between methods and studies are examined, and we conclude with some ideas for future research.

1. Introduction

Text classification is an old problem. As early as the 1800s, studies were done on verifying the authorship of the works of Shakespeare [1]. Since then, authorship identification has found use in an increasing number of fields, including criminology and security. It can be used for identification of frauds and other types of internet-based crime [2], as well as plagiarism detection and copyright disputes, checking the veracity of legal or historical documents, tracking criminals and terrorists, etc. The same knowledge can also be used in the reverse, to preserve anonymity on the Internet [3]. The methods have changed over time, from linguistic scholars pouring over documents, to purely statistical computer-based approaches. The field of text classification has also branched out, and similar methods are now being used for many other purposes than author identification. Studying the wide variety of methods available is important, not only for improving the accuracy, but also for measuring the accuracy, so that we know not to rely too much on insecure methods [1][4].

Still, after so many years of work, there is very little consensus on the best methods for any kind of classification [5]. One gets the impression that each new paper uses a completely new method, making it very difficult to compare the results. In this paper, we will look at some of the variables of text classification, and mention some of the things that have been done, and some of the things that remains to be done.

2. The process of classification

In general, automatic text classification uses a corpus, consisting of a number of texts with some target property we are interested in, such as the author or genre of the text. We extract some kind of features for each of the texts. Then we apply a mathematical model, a classifier, which somehow estimates the similarities between different texts based on the features, and guesses the target property. We shall look at some of the options in each of those steps.

2.1. **Type.** There are a few different variations of the classification problem in general. Most importantly, we distinguish between supervised and unsupervised classification. In supervised classification, we have a training set where the target property is known to the classifier, and a test set for which we want the classifier to give us the target property. When evaluating the system, we usually use a test set where the target property is known, so that we can see the accuracy of the classifier. In unsupervised classification, the classifier does not use any information about the target property of the texts, but tries to determine a set of classes or a similarity measure independently, based on the given features.

2.2. **Target.** One of the most important aspects of classification is author identification. This has many uses - for security purposes as well as commercial and personal use. We can also extend the search for documents by the same author to a search for authors who write in a similar style, for various purposes - finding literature recommendations, detecting influences of historical writers, or simply categorising authors or works.

But there are many other properties that we might want to detect. One is the topic of the text, something which is often done using simple keyword searches. Another is genre, such as determining whether a text is fact or fiction, poetry or prose.

Another category of properties are things we may want to know about the author. We can try to detect an author's gender, age, native language, dialect, etc.

There are also many more properties which are orthogonal to authorship. For example, many companies are interested in knowing whether reviews of their products are positive or negative, so we develop methods for analysing the mood of a text.

2.3. **Corpus.** Choosing the right corpus makes a big difference for author identification and text classification in general. To begin with, we generally get better results with a bigger corpus. It also matters what sort of information is given in the corpus. To successfully evaluate a method, we need to have at least one kind of metadata, namely the target property itself. There may also be other metadata we can use; if we are dealing with internet data, we may have information about the format or context, about medium-specific aspects such as font, and time the data was posted.

If we want to make a syntactic or semantic analysis of the text, we need to have that type of information; we need an annotated corpus, where for example the syntactical relations are written out for each word. Some corpuses have been manually analysed and annotated, but in most cases an annotated corpus has been produced by using a parser, a program which automatically performs the annotation. Parsers use partly the same methods as classifiers, and have partly the same problems. It can make a big difference what annotation scheme has been used, and how well the parser has performed.

There are many other issues to take into account when choosing a corpus. If we are attempting author identification, it matters how many authors are in the corpus, and how many texts are by each author - typically we want similar numbers of texts by each author. The length of each text can also make a big difference, and the homogeneity. We will probably get a higher accuracy if we use only texts written in the same context within a narrow field of topics, but that only works if we are planning to use the system on similar texts. If we train and test the system on one kind of text, and then apply it to a different kind, we may get an inflated idea of the accuracy. This means that if we know the texts we will apply the system to, we would prefer a corpus of similar texts; if the target texts are in a narrow field, we would prefer homogeneous training texts, but if the target texts are more diverse, or if we have no available corpus in the right field, we would prefer a more diverse training corpus.

2.4. **Features.** There is a wide array of features that can be used for classification; Rudman [5] finds that more than 1000 different style markers have been proposed. On an abstract level, we can divide features into simple features, which give one feature value per text, and feature compounds, which give several. They might of course also be of different data types, such as boolean or integer, but most of the interesting features are expressed as how-many-times-per-so-many-words, so they can be considered fractions, or (for texts of the same length) integers. The most common feature category is the lexical features. Here we find a few simple features, such as average lengths of words, sentences, paragraphs or texts, as well as a few complexity measures, including the number of different words used, and the number of words used only once. We can also imagine some compound versions, such as fractions of words with n letters. But perhaps the most important are the various versions of word frequencies. We might simply count the frequencies of the n most common tokens, or we might lemmatise them, that is, convert each word to its dictionary form before counting frequencies. The set of words we use can also be chosen in other ways. We may look at less common words, pairs of synonyms, misspelled words, only function words, or even use specific sets of words for each class we want to identify. Finally, we can also extend word frequencies to n-grams, that is, sequences of words. That further extends the number of possible features, and the need to choose a subset of them.

Another category which has become increasingly popular is character features. Here, we use essentially the same things as for the words, but apply them to characters instead [6].

If we have an annotated corpus, we can use syntactic and semantic features. As such, we might count simple things like part of speech frequencies, but usually the main focus in this category is to look at the frequencies of various patterns in the syntax tree.

2.5. **Classifier.** Most modern studies on text classification use one of a set of standard machine learning algorithms. This includes such diverse methods as decision trees, support vector machines, neural networks, and Bayesian classifiers.

3. Previous work

3.1. **Type.** Much of the early work in author identification was done manually. It was also often done on works where the author was actually unknown; rarely were the methods tested on known authors to verify the accuracy, so strictly speaking the studies were more speculation than real empirical science [1].

Much of the work done so far has focused on supervised learning, that is, the training corpus consists of texts by known authors. A smaller number of studies have looked at unsupervised learning, where the training corpus consists of texts with no author given.

In supervised learning, some methods take a profile-based approach, in that they treat all texts by an author as one, whereas others take an instance-based approach, treating the texts separately.

There are also, particularly for unsupervised learning, several questions we might be asking about author classification:
- general classification: which texts are by the same author?
- specific classification: which texts are by the same author as a given text?
- general similarity: how likely is each pair of texts to be by the same author?
- semi-specific similarity: how likely is each text to be by the same author as a given text?
- specific similarity: how likely is a given pair of texts to be by the same author?

3.2. **Target.** Much of the research done has been on author identification, a problem which despite great improvements in the past decades must be considered far from solved. The same methods that work for author identification have also been used to identify properties of authors, such as gender. This has proven to be feasible, but often more difficult than simple author classification [7]. Other text properties can be comparatively easy; Koppel et al. [8] tried using the same methods on gender identification and fiction / non-fiction identification, and achieved 80% and 98% accuracy, respectively.

Some simpler tasks have been solved quite well by basic machine learning algorithms. That includes language identification and spam detection, two types of classification with obvious real-life applications.

3.3. **Corpus.** Few studies have been done comparing the effects of different corpuses. In many cases, researchers have used multiple corpuses for comparison, specifically to rule out the influence of a specific corpus and be able to focus on the effects of choosing a particular feature set. A few studies have touched upon the effects of corpuses of different sizes, often from the perspective of comparing features or classifiers and seeing which perform best on smaller or larger datasets.

3.4. **Features.** Much has been said about word frequencies, and they remain one of the most commonly used features. As computational power increases, we also start looking at frequencies of n-grams, which can lead to very large feature sets. Lately, character n-grams have been increasingly popular, in many cases outperforming lexical features [6]. The most successful attempts at author identification often use diverse sets of features; Naranyan et al. (2012) [3] achieved an accuracy of 20% on 100 000 authors using a mixture of function word frequencies, syntax tree fragment frequencies, frequencies of special characters, and various complexity measures such as word lengths.

Most studies have used the same feature sets for all authors, but some have experimented with individual feature sets. Peng et al. (2003) used feature sets consisting of the top 5000 character n-grams for each writer, whereas Chaski (2001) looked at misspelled words, and formed individual feature sets based on the common misspellings of each writer.

Many have attempted to use various higher-level features, based on syntactic or semantic knowledge, with varying degrees of success. Moschitti and Basili [9] attempt to use noun phrases or word senses to improve the accuracy compared to just word frequencies, and find that the improvement is minute at best. Kim et al. [10] use a technique where several subtrees of the same tree is used as one feature, and found that it significantly improves the accuracy. Raghavan et al. [11] use a probabilistic grammar as model for the language of each author, and find that this works best in conjunction with simple lexical features. On the other hand, Chaski [4] tests several feature sets and finds that only syntactic analysis and syntactically classified punctuation are accurate enough to be reliable.

3.5. **Classifiers.** Support vector machines are perhaps the most popular classifier, but it quickly becomes overburdened when the number of features is large. This has lead to some work being done on how to combine the features to decrease the amount of work that the classifier needs to do. Another approach, which gained popularity in the 90s, is the naive Bayes method, which generally has much less trouble with large datasets and feature sets. As McCallum and Nigam [12] point out, there are a few different naive Bayes methods, which are rarely compared; they find the multinomial version to work better with large feature sets.

4. Future work

4.1. **Type.** There is definitely a need for more research on unsupervised classification. Particularly on the internet, we may not have a set of candidate authors.

There are also several types of semi-supervised classification. The traditional kind of semi-supervised machine learning occurs when a random subset of the data is labelled. This has been shown to be effective in other applications, but has not been studied much in text classification. Another type of particular interest would be when one author is known, but other texts are by an unknown set of authors. Finally, we may have a situation where some texts are suspected, but not known, to be by the same author.

Although both profile-based and instance-based methods have been tried before, there is little information on the differences between the results. It would be of interest to compare similar methods - the same features used on the same corpus - and see what the accuracy is for the profile-based and the instance-based approach.

4.2. **Target.** Author identification has been studied a lot, and needs to be studied more. But there are many other target properties that can be of interest. Among the simpler ones are some properties of the authors, like gender, age, native language, etc. Even when the categories are much fewer than the authors, it turns out that they can be much harder to identify, since the traits shared by a category might be fewer than those specific to an author. Another very important kind of target properties are those that are decidedly orthogonal to the author, such as genre, topic, or level of formality. Those features are important in themselves; one might for example want to restrict a database search to certain kinds of texts. But they are also crucial to success in author identification. Avoiding the influence of factors like topic is important both for successfully identifying an author in most real applications, and for accurately judging the accuracy of a method.

4.3. **Corpus.** The issue of author vs. topic is also important in choosing a corpus. In order to successfully explore author identification, one should ideally have a corpus containing a reasonable number of authors - with the authors known, which is far from always the case - and a very large number of texts by each author. For each author, we would like to have roughly the same number of texts, the same lengths of texts, and the same distribution of topics. Another issue is annotation. If syntactical methods are ever to be successful, it stands to reason that we need a well-parsed corpus. That is not to say that parsing errors necessarily ruin a classification; the parsing is still based on some properties of the text, and it is conceivable that an incorrect parsing might even give better results than the correct one would. But if the parsing is done by looking at simple features like word n-grams, then it may be that syntactical classification methods will be nothing but obscured lexical methods. In order for this to work, it would be helpful to have access to better hand-annotated corpuses, partly to directly test the theories of classification, and partly to improve on existing parsers. Sadly there are few large hand-annotated corpuses, and the ones that do exist are not based on the most suitable texts. For one thing, they might not contain the information we want, such as the name of the author. They might also be strangely unsuitable as training corpuses for parsers. One of the biggest existing hand-annotated corpuses is based on the New York Times, which is full of unusual words (technical language, proper nouns), strange syntax (for headers) and virtually unparsable text (such as lists and tables).

4.4. **Features.** So far, the history of text classification - and perhaps natural language processing in general - seems to be mostly victories for the ”stupid” methods. Methods using knowledge of syntax and semantics have proven to be generally less useful than simple methods like word frequencies. As mentioned above, part of this may be due to parsing errors in corpuses. On the other hand, seemingly even more inane features like letter frequencies have proven to be at least significantly useful. Why do we get better accuracy with such basic features? One likely reason is that they get much higher numbers, and classifiers generally perform better the more data they get. This is a field which seems to require much more research: What happens if the corpus grows very large? Clearly each method should improve, presumably eventually plateauing towards a maximum accuracy. One might conjecture that methods like character count, which are in all likelihood successful because they already have a lot of data, would reach their maximum sooner, and that the maximum would be lower. To test hypotheses like this, we would need a very large and suitable corpus, as mentioned above. Given enough information, we might expect at least word-level methods to outperform character-level methods. But as for syntax, it remains to be seen whether it can be of any real use. Perhaps the truth is that word n-grams successfully catch the syntactical information, as well as other information. Perhaps the main argument for using syntax is that it is independent of topic. This is also something that would require a suitable corpus to find out, in this case one with the right combinations of topics and authors. This is very difficult to find, partly because topic is of course not as clear-cut as author. There are also other parameters to take into account; some aspects of syntax may be independent of topic, but they are not independent of style [13]. It might be a difficult task to identify a person when he writes in a more or less formal context, for example.

One thing we might learn from cryptoanalysis is that frequencies of characters or words are often very different in different parts of a sentence. This information may be captured by n-grams (of words or characters), but even if we do not use the whole set of n-grams, it may be worth considering simply counting initial or final frequencies. For supervised techniques, the use of individual feature sets needs to be studied further. One simple approach would be to count the most common words for each author. Other options include looking at common words which the author does *not* use, or more generally finding those words where an author differs the most from the norm. That way we could perhaps get the advantages of a much larger feature set, without slowing down the process. There are also a number of other conceivable features which have not yet been tested. Parts of speech have been looked at, as well as pairs of parts of speech in the syntax tree, but perhaps pairs of one part of speech combined with one word would be of some interest. We could also make other divisions of word categories, for example into concrete and abstract nouns, etc. Since one of the important issues seems to be getting enough data, we could even go as far as to try to create variable-size feature sets, where categories are joined or split up depending on the size of the corpus. We would effectively get a "semantic tree" of words with varying degrees of similarity. There has been a fair amount of research into automatic classification of words, grouping words by similarity; this could perhaps come in handy for making such semantic tree features.

4.5. **Classifiers.** The choice of classifier still seems to be done quite haphazardly. One would assume that the choice of classifier depends on the feature set and corpus - perhaps SVM is more powerful for smaller feature sets, and Bayes for larger - but there is no real consensus, and other methods are still used. It might also be possible to fine-tune these methods, rather than blindly applying standard tools, perhaps using linguistic knowledge, or perhaps more likely using statistical knowledge.

It is worth noting that classifiers sometimes perform worse after adding more information [9]. Theoretically, one would imagine a more advanced classifier should should be able to identify features which are dragging it down and discard them.

Different classifiers also make various assumptions regarding the features. Naive Bayes makes, to begin with, the assumption that all features are uncorrelated - thus the "naive". This is clearly not true, but may still give good results. What the method does not assume is that there is any correlation between close values. For example, the training set might show that a text where 15 of the words are "you" is likely to be written by author A, and a text where 17 of the words are "you" is also likely to be written by author A. In this case, the NB method does not assume that a text where 16 of the words are "you" is also written by A.

SVM, on the other hand, might assume too much instead of too little in this regard. If we know that author A uses, on average, the word "and" slightly more often than the average, and we read a text where the word "and" is used *much* more than average, this might lead us to believe that this text is very likely to be by author A. This could perhaps be a problem if we are trying to identify just one author versus all others, as the others may be both higher and lower on a given feature.

4.6. **Conclusion.** Many of the studies done so far have attempted to increase the accuracy, to find the best features and set new records for how well we can identify authors, or whatever other property they are considering. Indeed, many studies specifically point out that they are devising a "novel approach" to the problem [10][11], rather than rather than replicating and verifying known techniques [5]. While this is of course very useful, it is perhaps not the most scientifically sound approach to the problem. The studies made typically vary in so many ways that it becomes impossible to compare them, and they themselves may not contain much of a comparison. It would be interesting to exhaustively compare all combinations of some feature sets, corpuses, and classifiers. Which features work well together, and which are redundant? Are different features better for different corpuses? Are different classifiers better for different features?

Perhaps as a result of the "stupid" methods prevalent in much of text classification, we now have a fairly good idea of what works and what doesn't, but we have very little idea why. Is the apparent failure of syntactical methods due to inaccuracies in parsing? If we have a corpus where we can successfully isolate author and topic, which methods will correlate with author and which will correlate with topic? And perhaps most importantly, what happens to the accuracy of the different features as the corpus size increases? Will the more advanced methods catch up and get an equivalent or better accuracy than the simple ones, or is it a fundamental property of writing that it is best recognised by low-level methods?

References

[1] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for Information Science and Technology*, vol. 60, 2009.

[2] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace." *ACM Transactions on Information Systems*, vol. 26.2, p. 7, 2008.

[3] A. Narayanan, H. Paskov, N. Zhenqiang Gong, J. Bethencourt, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *2012 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2012, pp. 300–314.

[4] C. E. Chaski, "Empirical evaluations of language-based author identification techniques." *Forensic Linguistics*, vol. 8, pp. 1–65, 2001.

[5] J. Rudman, "The state of authorship attribution studies: Some problems and solutions." *Computers and the Humanities*, vol. 31, pp. 351–365, 1998.

[6] F. Peng, D. Schuurmans, V. Keselj, and S. Wang, "Language independent authorship attribution using character level language models." in *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, 2003, pp. 267–274.

[7] N. Zechner, "Vocabulary and syntax in gender classification," in *Proceedings of the 4th Swedish Language Technology Conference*, 2012, pp. 81–82.

[8] M. Koppel, S. Argamon, and A. R. Shimoni, "Automatically categorizing written texts by author gender," *Literary and Linguistic Computing*, vol. 17, 2002.

[9] A. Moschitti and R. Basili, "Complex linguistic features for text classification: A comprehensive study," in *Advances in Information Retrieval*. Springer, 2004, pp. 181–196.

[10] S. Kim, H. Kim, T. Weninger, and J. Han, "Authorship classification: a syntactic tree mining approach," in *Proceedings of the ACM SIGKDD Workshop on Useful Patterns*, ser. UP '10. New York, NY, USA: ACM, 2010, pp. 65–73.

[11] S. Raghavan, A. Kovashka, and R. Mooney, "Authorship attribution using probabilistic context-free grammars," in *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 38–42.

[12] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification." *AAAI-98 workshop on learning for text categorization.*, vol. 752, pp. 41–48, 1998.

[13] C. Chung and J. Pennebaker, "The psychological functions of function words," in *Social Communication*, K. Fiedler, Ed. Psychology Press, 2007.