Studying Translationese at the Character Level

Marius Popescu
University of Bucharest
Department of Computer Science
Academiei 14, Bucharest, Romania
mpopescu@phobos.cs.unibuc.ro

Abstract

This paper presents a set of preliminary experiments which show that identifying translationese is possible with machine learning methods that work at character level, more precisely methods that use string kernels. But caution is necessary because string kernels very easily can introduce confounding factors.

1 Introduction

The term *translationese* designates the specific characteristics of translated texts compared to non translated text, the trace that the translation process leaves on translated texts. The term was introduced by Gellerstam in (1986). In an initial stage various features specific to translated texts, *universal features of translation*, were identified and corpus based approaches were employed to test the statistical significance of these translation universals (Baker, 1993; Laviosa, 2002). Recently, machine learning techniques have started to be used to investigate translationese: to distinguish between translated texts and non-translated ones, to identify the source language of a translated text, etc. (Baroni and Bernardini, 2006; Kurokawa et al., 2008; van Halteren, 2008; Ilisei et al., 2010; Koppel and Ordan, 2011).

These learning systems use a variety of features: (grammatical) words, part of speech tags, sentence length, etc.. By using this kind of features these methods implicitly handle the text at word level or above. Perhaps surprisingly, recent results have proved that methods that handle the text at character level can also be very effective in text analysis tasks. In (Lodhi et al., 2002) string kernels were used for document categorization with very good results. String kernels were also successfully used in authorship identification (Sanderson and Guenter, 2006; Popescu and Dinu, 2007) and plagiarism detection (Grozea et al., 2009).

In this paper we set to investigate if identifying translationese is possible with machine learning methods that work at the character level. More precisely, we will use string kernels in conjunction with different kernel methods in a series of experiments to see what performance can be achieved. Doing this we hope to answer the question if looking at the texts as just sequences of symbols (strings) is enough to identify translationese, and provide a method of identifying translationese that is language independent and theory neutral.

In the next section the related work and how our approach differs from it is discussed. In section 3 we briefly describe the kernel methods we used and string kernels. Section 4 describes the performed experiments and the obtained results, and the last section contains a discussion of these results and suggestions for future work.

2 Related Work

Using words is natural in text analysis tasks like text categorization (by topic), authorship identification and plagiarism detection. Perhaps surprisingly, recent results have proved that methods handling the text at character level can also be very effective in text analysis tasks. In (Lodhi et al., 2002) string kernels were used for document categorization with very good results. Trying to explain why treating documents as symbol sequences and using string kernels led to such good results the authors suppose that: “the [string] kernel is performing something similar to stemming, hence providing semantic links between words that the word kernel must view as distinct”. String kernels were also successfully used in authorship identification (Sanderson and Guenter, 2006; Popescu and Dinu, 2007). A possible reason for the success of string kernels in authorship identification is given in (Popescu and Dinu, 2007): “the similarity of two strings as it is measured by string kernels reflects the similarity of the two texts as it

Proceedings of Recent Advances in Natural Language Processing, pages 634–639,
Hissar, Bulgaria, 12-14 September 2011.

is given by the short words (2-5 characters) which usually are function words, but also takes into account other morphemes like suffixes ('ing' for example) which also can be good indicators of the authors style".

Even more interesting is the fact that two methods that obtained very good results for text categorization (by topic) (Lodhi et al., 2002) and authorship identification (Popescu and Dinu, 2007) are essentially the same, both are based on SVM and a string kernel of length 5. How is this possible? Traditionally, the two tasks, text categorization (by topic) and authorship identification are viewed as opposed. When words are used as features, for text categorization the (stemmed) content words are used (the stop words being eliminated), while for authorship identification the function words (stop words) are used as features, the others words (content words) being eliminated. Then, why did the same string kernel (of length 5) work well in both cases? In our opinion the key factor is the kernel-based learning algorithm. The string kernel implicitly embeds the texts in a high dimensional feature space, in our case the space of all (sub)strings of length 5. The kernel-based learning algorithm (SVM or another kernel method), aided by regularization, implicitly assigns a weight to each feature, thus selecting the features that are important for the discrimination task. In this way, in the case of text categorization the learning algorithm (SVM) enhances the features (substrings) representing stems of content words, while in the case of authorship identification the same learning algorithm enhances the features (substrings) representing function words.

Support Vector Machines (SVM) were also used in identifying translationese. Actually it is the dominant approach. In (Baroni and Bernardini, 2006) and (Kurokawa et al., 2008) the learning method used was SVM, In (van Halteren, 2008) and (Ilisei et al., 2010) several learning methods were used, SVM being included among them and reported to be among the top performers. Only in (Koppel and Ordan, 2011) a kernel method was not used. All these approaches use features computed at the word level or above: words, part of speech tags, sentence length, etc.. It might appear, because of the linguistically shallow representation, that these methods are language independent, but they directly or indirectly depend on resources specific to a given language. Most of the methods use part of speech tags (directly as features or indirectly as the proportion of some specific POS tags in text) and this implies the existence of a POS tagger for that language which is not always available. Even a method that uses as features only function words like the one in (Koppel and Ordan, 2011) is not completely language independent because it needs a way to segment a text into words which is not an easy task for some languages, like Chinese.

Using string kernels will make the corresponding learning method completely language independent because the texts will be treated as sequences of symbols (strings). Such a method will also have the advantage of being theory neutral. Methods working at the word level or above very often restrict their feature space according to theoretical or empirical principles. For example, they select only features that reflect *simplification universal* (Ilisei et al., 2010) or only some type of words (function words) (Koppel and Ordan, 2011), etc.. These features prove to be very effective for specific tasks, but other, possibly good features, depending on the particular task, may exist, for example source language specific features (Koppel and Ordan, 2011). String kernels embed the texts in a very large feature space (all substrings of length k) and leave it to the learning algorithm (SVM) to select important features for the specific task, by highly weighting these features.

3 Kernel Methods and String Kernels

Kernel-based learning algorithms work by embedding the data into a feature space (a Hilbert space), and searching for linear relations in that space. The embedding is performed implicitly, that is by specifying the inner product between each pair of points rather than by giving their coordinates explicitly.

Given an input set $\mathcal{X}$ (the space of examples), and an embedding vector space $\mathcal{F}$ (feature space), let $\phi : \mathcal{X} \to \mathcal{F}$ be an embedding map called feature map.

A *kernel* is a function k, such that for all $x, z \in \mathcal{X}$, $k(x,z) =< \phi(x), \phi(z) >$, where $< .,. >$ denotes the inner product in $\mathcal{F}$.

In the case of binary classification problems, kernel-based learning algorithms look for a discriminant function, a function that assigns $+1$ to examples belonging to one class and -1 to examples belonging to the other class. This function will be a linear function in the space $\mathcal{F}$, that means it will have the form:

$$f(x) = \text{sign}(< w, \phi(x) > +b),$$

for some weight vector w. The kernel can be exploited whenever the weight vector can be expressed as a linear combination of the training points, $\sum_{i=1}^{n} \alpha_i \phi(x_i)$, implying that f can be expressed as follows:

$$f(x) = \text{sign}(\sum_{i=1}^{n} \alpha_i k(x_i, x) + b)$$

.

Various kernel methods differ by the way in which they find the vector w (or equivalently the vector α). Support Vector Machines (SVM) try to find the vector w that defines the hyperplane that maximally separates the images in $\mathcal{F}$ of the training examples belonging to the two classes. Mathematically, SVMs choose the w and the b that satisfy the following optimization criterion:

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} [1 - y_i(< w, \phi(x_i) > +b)]_+ + \nu ||w||^2$$

where y_i is the label ($+1/-1$) of the training example x_i, ν a regularization parameter and $[x]_+ = \max(x, 0)$.

Kernel Fisher Discriminant (KFD) selects the w that gives the direction on which the training examples should be projected in order to obtain a maximum separation between the means of the two classes scaled according to the variances of the two classes in that direction. The optimization criterion is:

$$\max_{w} \frac{(\mu_w^+ - \mu_w^-)^2}{(\sigma_w^+)^2 + (\sigma_w^-)^2 + \lambda ||w||^2}$$

where μ_w^+ is the mean of the projection of positive examples onto the direction w, μ_w^- is the mean for the negative examples, σ_w^+ and σ_w^- are the corresponding standard deviations and λ is a regularization parameter. Details about SVM and KFD can be found in (Taylor and Cristianini, 2004). What is important is that the above optimization problems are solved in such a way that the coordinates of the embedded points are not needed, only their pairwise inner products, which in turn are given by the kernel function k, are required.

The kernel function offers to the kernel methods the power to naturally handle input data that are not in the form of numerical vectors, for example strings. The kernel function captures the intuitive notion of similarity between objects in a specific domain and can be any function defined on the respective domain that is symmetric and positive definite. For strings, many such kernel functions exist with various applications in computational biology and computational linguistics (Taylor and Cristianini, 2004).

Perhaps one of the most natural ways to measure the similarity of two strings is to count how many substrings of length p the two strings have in common. This gives rise to the p-spectrum kernel. Formally, for two strings over an alphabet Σ, $s, t \in \Sigma^*$, the p-spectrum kernel is defined as:

$$k_p(s, t) = \sum_{v \in \Sigma^p} \text{num}_v(s) \text{num}_v(t)$$

where $\text{num}_v(s)$ is the number of occurrences of string v as a substring in s [1]. The feature map defined by this kernel associates to each string a vector of dimension $|\Sigma|^p$ containing the histogram of frequencies of all its substrings of length p. Taking into account all substrings of length less than p, a kernel that is called the *blended spectrum kernel* will be obtained:

$$k_1^p(s, t) = \sum_{q=1}^{p} k_q(s, t)$$

The p-spectrum kernel will be the kernel that we shall be using in conjunction with SVM and KFD in our experiments. More precisely we shall use a normalized version of the kernel to allow a fair comparison of strings of different lengths:

$$\hat{k}_p(s, t) = \frac{k_p(s, t)}{\sqrt{k_p(s, s) k_p(t, t)}}$$

4 Evaluation

4.1 The Corpus

For our experiments we have assembled a corpus of literary works, most of them dating from the nineteenth century, with very few dating from the end of the eighteenth century or the beginning of twentieth century. All of them are book-length, the majority are novels, but also some essays, memoirs or autobiographies are included. In total we

[1] Note that the notion of substring requires contiguity. See (Taylor and Cristianini, 2004) for a discussion about the ambiguity between the terms ”substring” and ”subsequence” across different traditions: biology, computer science.

have collected 214 books. Half of them, 108, were originally written in English by both American and British authors. The other half of the corpus consists of translated works: 76 from French authors and 30 from German authors. The type of works we collected is very diverse, from classical works (works of Goethe, Balzac, Dickens) to popular fiction of the time (Eugène Sue's The Wandering Jew, the works of Karl May, Reynolds's Mysteries of London).

The source of the books was the Project Gutenberg[2], but in rare cases we also used other sources[3]. The Project Gutenberg policy - "We carry high quality ebooks: Our ebooks were previously published by *bona fide* publishers..." - ensures at least a minimal quality of the translated texts.

There is no space to list here all the titles in our corpus. Instead, in Table 1, we enumerate the authors represented in the corpus and the number of books by each author contained in the corpus.

Group	**Authors**
French authors	Balzac(10), Paul Bourget(1), Alphonse Daudet(3), Alexandre Dumas père(9), Alexandre Dumas fils(1), Flaubert(6), Anatole France(4), Théophile Gautier(1), Hugo(3), Hector Malot(2), Maupassant(6), Henry Murger(1), Prosper Mérimée(1), George Sand(1), Count Philip de Segur(1), Nahum Slouschz(1), Eugène Sue(1), Alexis de Tocqueville(1), Jules Verne(14), Émile Zola(9)
German authors	Berthold Auerbach(10), Gustav Freytag(1), E. T. A. Hoffmann(2), Goethe(2), Brothers Grimm(1), Friedrich Maximilian von Klinger(1), Karl May(5), Albert Pfister(1), Wilhelm Raabe(1), Leopold von Sacher-Masoch(1), Christoph von Schmid(1), Theodor W. Storm(1), Johann Ludwig Tieck(3)
American authors	James Fenimore Cooper(4), Stephen Crane(2), Nathaniel Hawthorne(4), Henry James(4), Herman Melville(4)
British authors	Jane Austen(5), Emily Brontë(1), Anne Brontë(2), Charlotte Brontë(4), George Bulwer-Lytton(4), Lewis Carroll(3), William Wilkie Collins(3), Joseph Conrad(4), Charles Dickens(13), Sir Arthur Coman Doyle(2), George Eliot(4), H. Rider Haggard(2), Thomas Hardy(5), George Reynolds(2), Sir Walter Scott(23), William Makepeace Thackeray(5), Anthony Trollope(5), H. G. Wells(3)

Table 1: The list of authors and the number of their books contained in the corpus

4.2 Experimental Setup

In all our experiments the objective was to learn a classifier able to distinguish translated texts from non-translated ones, thus obtaining a binary classification problem. The texts in the corpus were labeled as translated (T) if they were works of French and German authors translated into English or were labeled as original English (O) if they were works originally written in English by British or American authors.

Because the string kernels work at the character level, we didn't need to split the texts into words or to do any preprocessing. The only editing done to the texts was the replacing of sequences of consecutive space characters (space, tab new line, etc.) with only one space character. This normalization was needed in order to not artificially increase or decrease the similarity between texts because of different spacing.

In all experiments we have used a p-spectrum normalized kernel of length 5 ($\hat{k}_5$). We chose the length 5 to see if the same kernel that was reported to work well in the case of document categorization (Lodhi et al., 2002) and authorship identification (Popescu and Dinu, 2007) will also work for translationese identification. We did not attempt to optimize the value of the length of the p-spectrum kernel.

In all experiments the results obtained by KFD and SVM were almost identical. Here we reported only the result obtained by SVM.

p-spectrum kernel implicitly embeds the texts in a high dimensional feature space. Because we have a small number of examples (214), in a high dimensional feature space, the data set is separable and the best working SVM is a hard margin SVM that can be obtained setting the C parameter of the SVM to a high value (Taylor and Cristianini, 2004). In all our experiments the value of C was set to 100.

4.3 Experiments and Results

In the first experiment we have performed a cross-validation on the entire corpus. The goal of the cross-validation was not to set or tune any parameter of the learning method (all parameters were set by other criteria, see the previous section). The purpose of the cross-validation was to obtain a first estimation of the accuracy of the classifier learned by SVM based on a p-spectrum kernel of length 5. The 10-fold cross-validation accuracy was 99.53% and the leave one out cross-validation accuracy was 100%. The obtaine results are higher than the ones reported in other studies. In fact, the results were so good they made us suspicious.

In the second experiment we have prepared a much harder setting to test the learning method.

[2] http://www.gutenberg.org

[3] For example, the works of Karl May were taken from: http://www.karl-may-gesellschaft.de

We have used for training all the texts translated from French and the original English texts written by British authors. We have tested the obtained classifier on the texts translated from German and the original English texts written by American authors. This scenario is more difficult because training texts for the class T were translations from some fixed source language (French), while all test texts in T were translations from a different source language (German). Similar cases are discussed in (Koppel and Ordan, 2011). This setting also violates one of the fundamental assumptions in machine learning: that the training and test data are drawn from the same distribution. The accuracy obtained in this setting was 45.83%. This means that nothing was learned and the obtained classifier is a random one.

The great discrepancy between cross-validation accuracy and the accuracy obtained in the second experiment is a clear symptom of over-fitting. Most probable, the learning method found some features (substrings) that can be used to distinguish with very high accuracy between translated and non-translated texts in the case of training data, but failed to do the same thing in the case of test data. Because we have used a kernel method it is hard to examine individual features in order to see their importance within the classification function. But because we know the difference between training data and test data (the different source languages of the translated texts) we can guess what kind of features can act as confounding variables. Most likely these are substrings extracted from foreign proper names. Such substrings that differ from typical English substrings can be very good indicators of translationese. In the case of cross-validation typical French and German substrings can be seen in the training examples and this explains the good results. In the second experiment the learning method sees only French translations and thus fails to recognize German translations as translated texts.

One possible remedy to this problem would be to replace all proper names with a special string or symbol, solution adopted by others (Baroni and Bernardini, 2006) as well. But this would mean that our method treats texts at word level and not at character level. We opted for a more direct approach.

For the third experiment we have collected the French original of all the works of French authors in the corpus. These French texts formed a reference corpus. We modified the p-spectrum kernel so as to exclude all substrings that appear in the reference corpus. More precisely, when the p-spectrum kernel is computed between two texts, if a substring of length p that is common to the two texts is found, it will be counted as a common substring only if it does not appear as a substring of a text in the reference corpus. In this way, the substrings belonging to French proper names in the corpus will be eliminated from the feature space, but, of course, many other substrings will also be eliminated. This procedure was applied when the kernel was computed between any pair of texts from the corpus regardless of the source language (translated from French, translated from German or English original).

When we have repeated the previous experiment, training on texts from French and British authors and testing on texts from German and American authors, with the new kernel, the obtained accuracy was 77.08%. In a similar experiment, training using translated texts from French and testing using translated texts from German, but on a different data set (Europarl corpus), Koppel and Ordan (2011) reported an accuracy of 68.5%.

This third experiment proves that identifying translationese is possible with machine learning methods that work at the character level, using SVM and a modified string kernel.

Finally, we have performed a fourth experiment to see if the fact that we have used for training the works of British authors and for testing the works of American authors had any consequence regarding accuracy. As in the previous experiments we used for training translated texts from French authors and for testing translated texts from German authors. The original English texts, regardless of the nationality of the author, were randomly partitioned into 6 parts, one part being kept for testing and the other 5 being used for training (like in cross-validation, with the difference that the procedure was followed only for original English texts). The average accuracy obtained was 76.88%, being not significantly different from the accuracy obtained in the previous experiment (77.08%).

5 Conclusions and Further Work

In this paper we have performed a set of experiments regarding the identification of translationese using string kernel in conjunction with kernel methods. We have found that identifying translationese is possible with machine learning methods that work at the character level, SVM and string kernels, but caution is necessary because string kernels very easily can introduce confounding factors.

More experiments are needed in order to clarify all aspects of identifying translationese at the character level.

To eliminate the confounding factors introduced by p-spectrum kernel when training examples come from one source language and testing examples from another we used the original version of the translated texts in the training set. This is a strong requirement. Can we use as reference corpus other texts in the source language, not necessarily the original version of the translated texts? We plan to investigate this question.

It is likely that the confounding factors will not appear if a corpus more suited for studying translationese (comparable corpora (Laviosa, 1997)) will be used. We plan to test the method on such corpora (like Europarl).

Acknowledgments

Work supported by the National University Research Council of Romania (the "Ideas" research programme, PN II-IDEI), Contract No. 659/2009.

References

M. Baker. 1993. Corpus linguistics and translation studies, implications and applications. In M. Baker, editor, *Text and Technology, In honour of John Sinclair.* John Benjamins Publishing Company., Philadelphia/Amsterdam.

M. Baroni and S. Bernardini. 2006. A new approach to the study of translationese: Machine-learning the difference between original and translated text. *Literary and Linguistic Computing*, 21(3):259–274.

M. Gellerstam. 1986. Translationese in swedish novels translated from english. In L. Wollin and H. Lindqvist, editors, *Translation studies in Scandinavia*.

C. Grozea, C. Gehl, and M. Popescu. 2009. ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In *3rd PAN WORKSHOP. UNCOVERING PLAGIARISM, AUTHORSHIP AND SOCIAL SOFTWARE MISUSE*, page 10.

I. Ilisei, D. Inkpen, G. Corpas Pastor, and R. Mitkov. 2010. Identification of translationese: A machine learning approach. In A. F. Gelbukh, editor, *CICLing*, volume 6008 of *Lecture Notes in Computer Science*, pages 503–511. Springer.

M. Koppel and N. Ordan. 2011. Translationese and its dialects. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistic: Human Language Technologiess*, Portland, Oregon, USA, June. Association for Computational Linguistics.

D. Kurokawa, C. Goutte, and P. Isabelle. 2008. Automatic detection of translated text and its impact on machine translation. In *Proceedings of MT-Summit XII*.

S. Laviosa. 1997. How comparable can 'comparable corpora' be? *Targe*, 9(2):289–320.

S. Laviosa. 2002. *Corpus-based Translation Studies. Theory, Findings, Applications*. Amsterdam & New York: Rodopi.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Christopher J. C. H. Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

Marius Popescu and Liviu P. Dinu. 2007. Kernel methods and string kernels for authorship identification: The federalist papers case. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-07)*, Borovets, Bulgaria, September.

Conrad Sanderson and Simon Guenter. 2006. Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 482–491, Sydney, Australia, July. Association for Computational Linguistics.

J. S. Taylor and N. Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA.

H. van Halteren. 2008. Source language markers in EUROPARL translations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 937–944, Manchester, UK, August. Coling 2008 Organizing Committee.