The Influence of Text Pre-processing on Plagiarism Detection

Zdenek Ceska
Computer Science and Engineering
Faculty of Applied Sciences
University of West Bohemia
Pilsen, 306 14, Czech Republic
zceska@kiv.zcu.cz

Chris Fox
School of Computer Science
and Electronic Engineering
University of Essex
Colchester, CO4 3SQ, United Kingdom
foxcj@essex.ac.uk

Abstract

This paper explores the influence of text pre-processing techniques on plagiarism detection. We examine stop-word removal, lemmatization, number replacement, synonymy recognition, and word generalization. We also look into the influence of punctuation and word-order within N-grams. All these techniques are evaluated according to their impact on F_1-measure and speed of execution. Our experiments were performed on a Czech corpus of plagiarized documents about politics. At the end of this paper, we propose what we consider to be the best combination of text pre-processing techniques.

Keywords

Plagiarism, Copy Detection, Natural Language Processing, Stop-words, Lemmatization, Synonymy, WordNet, Thesaurus.

1 Introduction

Recently, there has been much interest in automatic plagiarism. Written-text plagiarism is a wide-spread problem which many organizations have to deal with. Various methods are used in this field, such as SCAM [9] and Kopi [5]. SVDPlag [1] is another technique whose performance outperforms all these other methods.

Text pre-processing can have a significant influence on the performance of many Natural Language Processing (NLP) tasks, including plagiarism detection. Although many studies on pre-processing techniques have been performed for applications such as text categorization [10], it is appropriate to look at such pre-processing techniques again when when considering a new application. Plagiarism detection is a distinct field that should be given particular attention, as it may be appropriate to apply a wide range of pre-processing techniques. Various pre-processing have different effects, some improve the accuracy, some just decrease the time requirements, and some do both. This paper aims to clarify the influence of text pre-processing on this task when using the SVDPlag method.

2 Pre-processing Techniques

Plagiarism detection can employ various pre-processing techniques in order to improve the accuracy or decrease the number of features that need to be processed.

Figure 1 shows the text pre-processing step-by-step. The most essential block is Tokenization, which extracts single words from the structured text. Punctuation marks can be extracted if they are required by other processes. The other blocks represent optional techniques that can be applied if the user wishes.

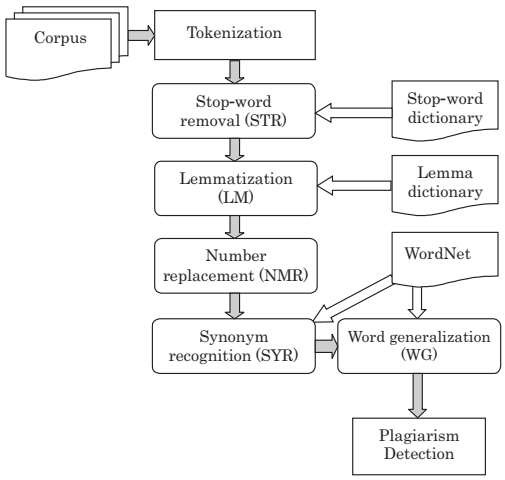

Fig. 1: *Text pre-processing scheme*

Below we describe each technique in detail. The impact of these techniques on accuracy and processing time is given in Section 4, where we also explore the impact of maintaining word-order, and the boundaries between sentences and phrases, as marked by punctuation.

2.1 Stop-word Removal (STR)

Stop-word removal is a fundamental pre-processing approach that removes common words. Its primary use is to prevent the following processing being over-influenced by very frequent words.

For plagiarism detection, there may be a complication to remove such words that could break up an

International Conference RANLP 2009 - Borovets, Bulgaria, pages 55–59

author's writing style. For this reason, the effect of stop-word removal is rather unpredictable. The usual way of determining what counts as a stop-word is just to use a dictionary that lists them. We used a pre-existing list for Czech [8].

2.2 Lemmatization (LM)

Lemmatization is the process of determining the base form of a given word [4]. During this process, the context of the word is used to determine the word sense.

Sometimes lemmatization is mistaken for stemming; however, there is an essential difference. Stemming operates only with single words without any knowledge of the context, and therefore cannot distinguish among words having several different meanings. As an example of stemming, the words "does" and "done" may be transformed into the stem "do". The resulting word does not need to be a real English word.

Lemmatization, on the other hand, makes use of the context to disambiguate word meaning. This is particularly important for languages that have rich systems of inflexion, such as Czech. We employed a method by Toman [10].

2.3 Number Replacement (NMR)

Number replacement is a particular approach that transforms all numbers into a dummy symbol. We suggest this approach may be highly appropriate in some cases. Let us imagine the situation when a student submits a plagiarized essay that focuses on an economic analysis of a company. It is very simple to use someone else's work just by replacing any numeric values, in combination with any necessary rewording. On the other hand, the number replacement could lead to lower accuracy in the case of factual dates.

2.4 Synonymy Recognition (SYR)

The motivation for using synonymy recognition comes from considering human behaviour, whereby people may seek to hide plagiarism by replacing words with appropriate synonyms.

If a sufficient number of words are replaced by synonyms, then most of the common copy detection methods fail. Regardless of the features the methods use, the best solution is to transform words having the same meaning onto a unique identifier. A consideration has to be given to words that have more than one meaning; if a significant impact on the accuracy is expected, a disambiguation process is required to determine the appropriate meaning.

We consider three possible solutions for synonymy recognition. These all exploit the WordNet thesaurus [12]. In WordNet, all words that have the same meaning are grouped together into a so-called *synset*. Moreover, each WordNet synset is mapping onto an *inter-lingual index* (ILI) that is used as a unique identifier.

2.4.1 First Meaning Selection (FMS)

To implement *first meaning selection* we search for an equivalent word in WordNet. If a match is found, then the algorithm returns the corresponding ILI. This approach does not care about ambiguity; even if there is more than one meaning for the word, it still just returns the first ILI.

2.4.2 Disambiguation and Proper Meaning Selection (DPMS)

A more advanced approach is to use a disambiguation process based on a Naïve Bayes classifier [6]. This aims to select the best word meaning depending on the adjacent words. For more information about the disambiguation process see [4]. Because the adjacent words do not always provide sufficient information for full disambiguation, this process sometimes fails.

2.4.3 Every Meaning Selection (EMS)

The last approach is a generalized variant of FMS. This selects all corresponding meanings contained in WordNet and returns their ILIs. For two words to be matched, at least one of their possible meanings has to correspond. There is a potential risk that this it too permissive.

2.5 Word Generalization (WG)

The last technique is word generalization. The main idea of this process rests in replacing various specific words by a more general word. For example, the words "dog" and "cat" could both be replaced by the word "animal", or some other hypernym, such as "mammal". This has two aims. First, it reduces the number of distinct words that have to be processed. Second, it may reveal evidence of plagiarism where some paraphrasing and generalization, or specialization, has been used in an attempt to hide the offence.

The WordNet thesaurus interconnects synsets by many *inter-lingual references* (ILR), where a synset consists of one or more synonyms. The hypernym relation defines a synset hierarchy.

The idea of replacing specific words by more general words is simple. The issue we have to address is how to decide which words to use. If we are insufficiently general, then there may be little benefit. If too general, then all nominals would be replaced by "entity", thus eliminating the information content.

The best solution might be to define an individual generalization level for every sub-hierarchy. However, this is rather impractical. We adopt the more practical alternative of specifying a fixed, global generalization level. All words from deeper, more specific levels of the hierarchy are replaced by the word occurring at that level. Words that are associated with shallower, more general levels are left unchanged.

Although we have been talking about replacing words with more general ones, in practice all that is required is to record the appropriate ILI of the relevant synset to which a given word belongs. All words have to be mapped onto their ILIs before the word generalization process. It turns out that this notion of word generalization is closely connected with the synonym recognition (SYR).

3 Plagiarism Detection Method

All the following experiments were performed using a variant of the SVDPlag method published in [1]. We modified this method in order to improve the evaluation when the documents are of differing length. The modification rests in an asymmetric document similarity normalization (Formula (1)). The modified method is called SVDPlag$_{\text{ASYM}}$. The original method (which we shall call SVDPlag$_{\text{SYM}}$) used symmetric normalization.

$$\text{sim}_{\text{ASYM}}(R,S) = \text{sim}_{\text{SVD}}(R,S) \cdot \frac{\sqrt{|G_{red}(R)| \cdot |G_{red}(S)|}}{\min(|G_{orig}(R)|,\ |G_{orig}(S)|)} \quad (1)$$

In this formula, $\text{sim}_{\text{ASYM}}(R,S)$ represents the resulting similarity between documents R and S. The term $\text{sim}_{\text{SVD}}(R,S)$ is a similarity measure given by the SVD process [1], $G_{orig}(D)$ denotes a set of N-grams contained in document D before reduction, and $G_{red}(D)$ is a set of N-grams after reduction.

4 Experiments

For our experiments we used a corpus of plagiarized documents, written in Czech. In total, the corpus contains 1,500 text documents about politics. The corpus was created as follows. Initially, 350 documents were selected from the Czech News Agency (CTK) source [3], volume 1999. A group of students was then set the task of manually plagiarizing these documents to create 550 plagiarized texts. A further 600 documents from CTK on the same topic were then added to the corpus. These documents effectively act as an unplagiarized control.

In order to produce the 550 plagiarized documents, students were asked to combine two or more randomly selected documents from the initial corpus of 350 CTK documents. During this task the following rules had to be taken into account: (i) copy several paragraphs from the selected documents; (ii) remove about 20% of sentences from the new created document; (iii) remove about 10% of words with consideration of the sentence meaning; (iv) exchange about 20% of sentences from different paragraphs; (v) modify some words or reword at most 10% of sentences to add new ideas; (vi) insert new words to fix any "broken" meanings.

To evaluate various pre-processing techniques the standard measures from Information Retrieval (IR) are used. We define precision p and recall r according to Rijsbergen [11]. Further we define F_1-measure to be a harmonic mean of precision and recall, see the following formula.

$$F_1 = \frac{2 \cdot p \cdot r}{p + r} \quad (2)$$

To make a comparison of time requirements, all the following experiments were performed on Intel Core 2 Duo E6600, 4GB RAM, and Windows Server 2003 R2 operating system in 64-bit mode.

Through all the experiments, the Student's t-test of significance at the confidence level of 99.5% was employed.

4.1 Punctuation and Word-Order

In the first experiment we look at the influence of considering punctuation and word-order on the accuracy, see Table 1. At the same time, we determine which features (N-grams) achieve the best results. We performed all the experiments on the asymmetric variant of SVDPlag.

We use the term "punctuation" to refer to the case where N-grams are ignored if they cross sentential and phrase boundaries, as marked by punctuation such as full-stops, question marks, commas, etc.

Table 1: *The influence of punctuation and word-order on the plagiarism detection accuracy*

Punctuation	–	+	–	+
Word-order	–	–	+	+
Features	F_1 [%]	F_1 [%]	F_1 [%]	F_1 [%]
Words	86.29	86.29	86.29	86.29
Bigrams	91.93	91.74	92.03	91.85
Trigrams	94.59	93.16	94.50	93.21
4-grams	95.68	93.23	95.56	93.33
5-grams	94.64	92.18	94.48	92.15
6-grams	93.16	90.29	93.07	90.33
7-grams	92.05	88.16	92.00	88.21
8-grams	90.54	85.69	90.41	85.58

¿From our experiments it is evident that people usually copy segments of text that include more than one sentence, or phrase. The F_1-measure decreases when N-grams that link sentences and phrases that may have been copied together are ignored. Moreover, the longer N-gram the larger the decrease in F_1-measure. Short sentences that have fewer words than the specified N-gram length are left out of computation process. Overall, this has the effect of reducing the number of features extracted from the text, which has the advantage of speeding-up the subsequent computation. Although the computation time for the SVD method decreases, the pre-processing takes more time due to the need to examine and process the sentential structure of the documents. Overall, just a few seconds are saved by applying this approach (Table 2).

Table 2: *The number of 4-gram features and the time requirements when punctuation and/or word order is taking into account*

Punctuation	Word-order	Num. of features	Preproc. time [s]	Comp. time [s]	Total time [s]
–	–	240159	23.65	60.73	84.38
+	–	155871	32.92	44.63	77.54
–	+	238903	23.48	60.33	83.81
+	+	155092	32.84	44.61	77.45

Next, we examine the impact of word-order by comparing the case where the order of the words within each N-gram is preserved with one where the words are sorted into ascending alphabetic order, effectively ignoring the original word order. This can be thought of as using an N-bag of words, where word-order is unimportant. It turns out that ignoring word-order gives slightly worse results (Table 2).

The number of features, and the time taken to process them, are reduced. Despite that, we do not recommend these techniques, especially with languages that do not support free word-order. All of the subsequent experiments were performed without considering punctuation, and with the word-order being maintained within the N-grams.

4.2 Pre-processing Techniques

Now we observe the influence of the individual pre-processing techniques, described in Section 2. Figure 2 presents the F_1-measure (the line) and the total computation time (the bars) for different pre-processing techniques, using 4-grams. The first case is where no pre-processing technique is applied. In this case, the F_1-measure is 95.68% and the total computation takes 84.38 seconds.

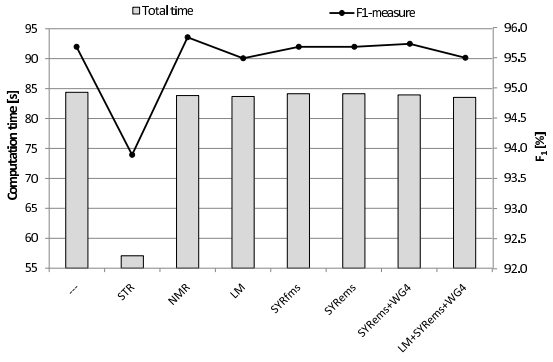

Fig. 2: *The influence of individual pre-processing techniques on F_1-measure and computation time, when* SVDPLAG$_{\text{ASYM}}$ *and 4-gram features are used*

Stop word removal (STR) significantly reduces the number of 4-grams and decreases time requirements to 57.08 seconds. Unfortunately, the F_1-measure falls down on 93.89%. This suggests that very frequent stop-words make a measurable contribution in determining the identity of fragments of text.

Number replacement (NMR) gives very promising result. A higher F_1-measure of 95.84% is obtained, together with a slightly lower execution time.

Lemmatization (LM) looses some relevant information; our experiments indicate a decrease in F_1-measure to 95.49%.

In the case of synonym replacement (SYR), we initially examine only the SYR$_{\text{FMS}}$ and SYR$_{\text{EMS}}$ approaches. This is because there is no training data for Czech word-meaning disambiguation. We discuss SYR$_{\text{DPMS}}$ later in Section 4.2.3. Both SYR$_{\text{FMS}}$ and SYR$_{\text{EMS}}$ have no influence F_1-measure. We just notice a small decrease in the time required. In the case of single words (Figure 3), there is a tiny improvement of just several hundredths of a percent, which is not significant. Generally, SYR$_{\text{EMS}}$ performs better than SYR$_{\text{FMS}}$.

These results suggest that in our data-set people often copy longer sentences without any modification. Nevertheless, sometimes a word is replaced by its synonym, which is reflected in the better results for single words. It seems the extra performance obtained by using N-grams (rather than single words) is not further improved by considering synonyms.

For word generalization (WG), we use the 4$^{\text{th}}$ generalization level (WG$_4$). As already mentioned in Section 2.5, WG cannot be applied by itself; it always has to be combined with a synonym replacement (SYR) technique. Word generalization has more impact on the computation time for shorter N-grams, and in particular for single words (see Section 4.2.1). However, we do also notice some improvement for 4-grams. It also gives a very slightly better results in F_1-measure (+0.05%) for both single words and 4-grams than single SYR techniques, although this may not be significant.

We explored the use of different generalization levels for WG. If the generalization level is too abstract, e.g. effectively replacing all words by "entity", then this has a negative impact on the F_1-measure. According to our experiments, the 4$^{\text{th}}$ generalization level achieves the best results, with the benefits gradually disappearing as greater levels of generalization are used.

4.2.1 Single Words

It is worth looking at the performance for single-word features, see Figure 3. Perhaps the most interesting result is the much smaller reduction in the F_1-measure and time requirements when STR is applied.

The Czech language has a rich system of word inflections, resulting in large reduction in the number of features, and overall computation time, when using LM with single words. Intuitively at least, it would seem that longer word sequence, e.g. 4-grams, contain implicit constraints on the occurrence of inflected forms of individual words, which reduces the impact of the LM technique.

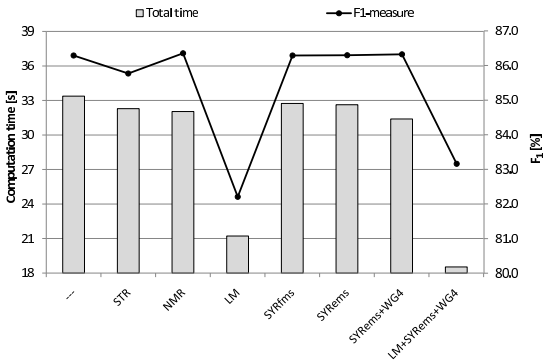

Fig. 3: *The influence of individual techniques on F_1-measure and the computation time, when* SVDPLAG$_{\text{ASYM}}$ *and single-word features are used*

The results for SYR and WG with single words were described above.

The last case for single words is the combination of LM, SYR$_{\text{EMS}}$ and WG$_4$. This combination resulted in an even greater reduction in time taken, and better results for F_1-measure.

4.2.2 Combinations of Techniques

It is possible to consider various combinations of text pre-processing techniques. Here there is only space to outline the results for a couple of combinations. The combination of NMR, SYR_{EMS} and WG_4 gives promising result; the F_1-measure is 95.92% in comparison with 95.68% when no pre-processing is used.

Using all the techniques together, i.e. STR, LM, NMR, SYR_{EMS}, and WG_4, yields a lower F_1-measure of 94.52% but is the best choice for obtaining a lower total execution time. In this case, using the combination of all the various techniques seems to gain the benefit of each one.

4.2.3 Sense Disambiguation (DPMS)

Finally, we examine SYR_{DPMS}, including the disambiguation process. Since there are no training data available for Czech language, we performed our experiments on the METER corpus [2]. This consists of news stories published in nine British newspapers, some of which are based on common news-wire sources.

For our experiments we used the $\text{SVDPLAG}_{\text{ASYM}}$ method with single-word features. Every piece of news is written in a novel style, which may explain why longer N-grams yield worse results. As a training corpus for English word disambiguation, we employed the Semantic Concordance Corpus [7].

According to our experiments, we achieve 87.07% F_1-measure without the use of pre-processing. Applying the SYR techniques slightly improves the results: 87.07% for FMS; 87.09% for DPMS, using a six word context centered on the word being disambiguated (DPMS-6); and 87.10% for EMS. According to the statistical significance testing, the measured differences are not significant. Using anything other than a six word context for DPMS gave F_1-measures that were worse than those without pre-processing.

The results suggest that DPMS has a worse performance than EMS. The reason for this appears to be that if a word is not recognized among the training data, a random meaning is selected. We would argue that EMS is good choice not only for this corpus, but also for the plagiarism detection problem in general.

5 Conclusion

On the basis of our experiments, text pre-processing cannot significantly improve the accuracy of plagiarism detection. Only number replacement (NMR), synonymy recognition (SYR), and word generalization (WG) improve the accuracy slightly. Their combination yields the highest score 95.92% F_1-measure in comparison with the situation when no pre-processing is employed, i.e. 95.68%.

If speed of execution is the main priority, then stop-word removal (STR) and lemmatization (LM) should be considered. Stop-word removal gives a greater reduction in execution time as longer N-grams are used. However, we should be aware that this throws away information that is useful for plagiarism detection, with the F_1-measure decreasing to 93.89%. Lemmatization (LM), on the other hand, has a greater impact on execution time with shorter N-grams. In case of single words, F_1-measure declines from 86.29% to 82.21%. We hypothesis that lemmatization has less impact because the word collocation contains implicit information about the legitimate inflexions.

Taking punctuation into account has a significant, but negative impact. Although it reduces the number of features that have to be analyzed, the overall execution time does not decrease. This is because of the additional time required to perform the pre-processing. The longer the N-grams, the greater is the reduction in the F_1-measure. An explanation for this is that short sentences do not fill the longer N-grams, and are simply discarded. Maintainingm or ignoring word-order has no influence on the results.

Synonymy recognition itself (SYR) does not improve the performance with longer N-grams. Out of all synonymy techniques, the approach that considers all of the word meanings, i.e. EMS, appears to have the best performance. Word generalization (WG) works in combination with the synonymy recognition and makes additional use of the WordNet thesaurus. According to our experiments, the 4^{th} generalization level achieves the best results for this technique.

Acknowledgments

This research was supported in part by National Research Programme II, project 2C06009 (COT-SEWing). Special thanks go to Michal Toman who helped us to employ the disambiguation process.

References

[1] Z. Ceska. Plagiarism Detection Based on Singular Value Decomposition. *Advances in Natural Language Processing*, 5221: 108–119, August 2008.

[2] P. Clough and S. Piao. The METER Corpus. Last update 12/2/2000. Available at `http://www.dcs.shef.ac.uk/nlp/meter/Metercorpus/metercorpus.htm`.

[3] CTK. Czech News Agency: Political and General News Service. Available at `http://www.ctk.eu/services/news/political_and_general/`.

[4] K. Jezek and M. Toman. Documents Categorization in Multilingual Environment. In *Proceedings of ElPub 2005*, pages 97–104, Leuven, Belgium, 2005. ISBN 90-429-1645-1.

[5] L. Kovacs and M. Pataki. KOPI Protection instead of Copy Protection. In *Proceedings of AXMEDIS 2006*, pages 38–41, Washington, DC, USA, 2006. ISBN 88-8453-526-3.

[6] C. Manning and H. Schutze. *Foundation of Statistical Natural Language Processing*. The MIT Press, Massachusetts Institute of Technology, Cambridge, USA, 1999. ISBN 0-262-13360-1.

[7] G. A. Miller, C. Leacock, R. Tengi, and R. T. Bunker. A Semantic Concordance. In *Proceedings of Human Language Technology Conference*, pages 303–308, Princeton, USA, 1993.

[8] RANKS.NL. Czech stopwords. Available at `http://www.ranks.nl/`.

[9] N. Shivakumar and H. Garcia-Molina. SCAM: A copy detection mechanism for digital documents. In *Proceedings of Digital Libraries '95*, pages 303–308, Austin, USA, 1995.

[10] M. Toman, R. Tesar, and K. Jezek. Influence of Word Normalization on Text Classification. In *Proceedings of InSciT 2006*, pages 354–358, Merida, Spain, 2006. ISBN 84-611-3105-3.

[11] C. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 2 edition, 1979. ISBN 0-408-70929-4.

[12] P. Vossen. Global WordNet Association: EuroWordNet. Last update 9/1/2001. Available at `http://www.illc.uva.nl/EuroWordNet/`.