Investigating Advanced Techniques for Document Content Similarity Applied to External Plagiarism Analysis

Daniel Micol and **Rafael Muñoz**
Dept. of Software and Computing Systems
University of Alicante
San Vicente del Raspeig, Alicante, Spain
{dmicol,rafael}@dlsi.ua.es

Óscar Ferrández
Dept. of Biomedical Informatics
University of Utah
Salt Lake City, Utah, USA
oscar.ferrandez@utah.edu

Abstract

We present an approach to perform external plagiarism analysis by applying several similarity detection techniques, such as lexical measures and a textual entailment recognition system developed by our research group. Some of the least expensive features of this system are applied to all corpus documents to detect those that are likely to be plagiarized. After this is done, the whole system is applied over this subset of documents to extract the exact n-grams that have been plagiarized, given that we now have less data to process and therefore can use a more complex and costly function. Apart from the application of strictly lexical measures, we also experiment with a textual entailment recognition system to detect plagiarisms with a high level of obfuscation. In addition, we experiment with the application of a spell corrector and a machine translation system to handle misspellings and plagiarisms translated into different languages, respectively.

1 Introduction

We believe there are two main user scenarios where external plagiarism detection tools are applied, sharing both of them the fact that they have a large source documents corpus. The difference, however, is that the first scenario is based on a large number of suspicious documents being processed at the same time, so the detection approach needs to be highly efficient and scalable. An example of this scenario would be the *1st and 2nd International Competitions on Plagiarism Detection* (Potthast et al., 2009; Potthast et al., 2010), where the corpora contain multiple source and suspicious documents. For this first use case we have developed a system to detect external document plagiarism that is highly efficient and scalable. It contains a first phase where a small subset of source documents are selected as possible candidates to be the origin of the plagiarism for a given suspicious document. Given that this phase processes the whole corpora, it uses a simple and lightweight function to select the subset of candidate source documents. After this is done, a more complex function is applied over this subset to extract which documents contain the plagiarism, and the exact position within these documents. This two-step approach is common among research systems, as described in (Potthast et al., 2009).

The second use case assumes that we only have to process one suspicious document at a time. Therefore, we can apply more complex techniques that are less efficient but highly accurate, as there is less data to process. An example of this use case could be an online system to detect if a scientific manuscript that an author wants to submit to a journal or conference is a plagiarism of a previously published paper. For this second use case we have experimented with more complex and accurate techniques, such as the usage of textual entailment recognition methods developed by our research group. In addition, we have also applied a spell corrector and a machine translation system to handle documents with misspellings and written in different languages.

2 State of the art

Most of the research approaches on external plagiarism analysis contain a simple and efficient heuristic retrieval to reduce the number of source documents to compare against, and a more complex and costly detailed analysis that attempts to extract the exact position of the plagiarized fragment, if any (Potthast et al., 2009). The system that we have developed is in line with this archi-

Proceedings of Recent Advances in Natural Language Processing, pages 240–246,
Hissar, Bulgaria, 12-14 September 2011.

tecture.

With regards to the heuristic retrieval, (Basile et al., 2008; Grozea et al., 2009) decided to apply a document similarity function that would be used as heuristic to determine if a given suspicious and source documents are similar enough to hold a plagiarism relation. (Kasprzak et al., 2009) create an inverted index of the corpus document's contents in order to be able to retrieve efficiently a set of documents that contain a set of n-grams. (Grozea et al., 2009; Stamatatos, 2009) implement a character-level n-gram comparison and apply a cosine similarity function based on term frequency weights. With this approach they extract the 51 most similar source documents to the suspicious one being analyzed. (Basile et al., 2009; Kasprzak et al., 2009) decided to implement a word-level n-gram comparison. Low granularity word n-grams, with a size of 1, have been explored by (Muhr et al., 2009), applying cosine similarity using frequency weights to extract the two most similar partitions for every sentence in a document, using the source document's sentences as centroid.

For the detailed analysis, (Basile et al., 2009) perform a greedy match merging if the distance of the matches is not too high. A more strict approach has been presented by (Muhr et al., 2009), requiring exact sentence matches, and afterwards applying a match merging approach by greedily joining consecutive sentences. In this method, gaps are allowed if the respective sentences are similar to the corresponding sentences in the other document. (Grozea et al., 2009) perform a computation of the distances of adjacent matches, joining them based on a Monte Carlo optimization. Afterwards, they propose a refinement of the obtained section pairs. (Kasprzak et al., 2009) extract matches of word n-grams of length 5, and apply a Match Merging Heuristic to get larger matches. Then they extract the maximum size that shares at least 20 matches, including the first and the last n-gram of the matching sections, and for which 2 adjacent matches are at most 49 not-matching n-grams apart.

3 Methods

We will first present a baseline system that is efficient and scalable, and designed to work for the first use case mentioned above. For this purpose, we will use corpora of thousands of suspicious and source documents, where every suspicious can contain none, one or more plagiarisms of any source documents. After this, we present certain optimizations built on top of our baseline system that will make it more accurate, although slower, and therefore will be applicable in the second use case.

3.1 Baseline system

Our baseline system (Micol et al., 2010), developed for our participation in the *2nd International Competition on Plagiarism Detection* (Potthast et al., 2010), has two phases: document selection, using a heuristic retrieval, and passage matching, performing a more detailed analysis.

The first step is to select a subset of candidate source documents that will later on be compared against a given suspicious document. This should reduce by a large factor the number of document comparisons to perform. To generate this set we will have to loop through all source documents, and given that this set is large, this operation needs to be relatively simple and inexpensive. Our approach to solve this problem is to weight the words in every document and then compare the weights of those terms that appear in both the suspicious and the source documents being compared. Their similarity score will be the sum of the mentioned common term weights.

Once we have a small subset of source documents to compare against for every suspicious one, we can perform a more accurate and costly comparison between pairs of documents. We try to find the largest common substring between suspicious and source documents, requiring a minimum length which will be the n-gram size. Once the n-grams of the source document being compared against have been extracted, we will iterate through the contents of the suspicious document, extract n-grams starting at every given offset, look them up in the list of n-grams of the aforementioned source document, and seek directly to the positions where the given n-gram appears, avoiding unnecessary comparisons. From these offsets we will try to find the largest common substring to both documents.

3.2 DLSITE: a textual entailment recognition system

The baseline system that we have detailed before is suitable for low levels of plagiarism obfuscation, given that it is based on lexical comparisons. If the person who performs the appropriation uses

equivalent terms instead of the original ones, or swaps the word order considerably, our system will not perform well and won't recognize these plagiarisms. To be able to detect these sorts of appropriations, we add semantic and syntactic techniques, as well as more advanced lexical measures.

Concretely, we decided to apply DLSITE (Ferrández et al., 2007a), a textual entailment recognition system developed by our research group that analyzes pairs of sentences, being one the text and the other the hypothesis, trying to determine if the hypothesis' meaning can be inferred from the text's. Therefore, with the use of this system, we could detect plagiarisms that are written in different manners, but still share their meaning. DLSITE contains the following modules:

Lexical analysis The lexical module of DLSITE (Ferrández et al., 2007b) computes the extraction of several lexical feature values for a given text-hypothesis pair. These measures are mainly based on word co-occurrences in both the hypothesis and the text, as well as the context where they appear.

Syntactic analysis The syntactic module of DLSITE (Micol et al., 2007) compares the meaning of the text and the hypothesis by generating their corresponding syntactic dependency trees, and then analyzing the similarities of these two structures. It is composed of a pipeline of four submodules, which are syntactic dependency tree construction, filtering, embedded subtree search and graph node matching.

Semantic analysis The semantic module of DLSITE analyzes a text-hypothesis pair from a meaning's perspective, using resources such as WordNet, VerbOcean and FrameNet. Similar research projects have already developed procedures using standard WordNet-based similarities (Corley and Mihalcea, 2005; Hickl and Bensley, 2007). However, in our case we also consider string-based similarities for the final similarity score. This allows us to positively consider entities that, while not appearing in WordNet, are very relevant, instead of penalizing their similarity score. We exploit WordNet relations in order to find semantic paths that connect two concepts through the WordNet taxonomy.

Since verbs have a strong contribution to the sentence's final meaning, we want to measure how the hypothesis' verbs are related to the text's. To achieve this, we exploit the VerbNet lexicon (Kipper et al., 2006), and the VerbOcean and WordNet relationships, trying to find correlations between the main verbs expressed in the hypothesis with those in the text. The underlying intuition about the VerbNet correspondence is that the verbs wrapped in the same VerbNet class or in one of their subclasses have a strong semantic relation since they share the same thematic roles and restrictions, as well as syntactic and semantic frames. Additionally, VerbOcean's relations are good indicators of semantic correspondence between verbs.

Another relevant issue to recognize entailment relations is to analyze the presence and absence of named entities. (Rodrigo et al., 2008) successfully built their system mainly using the knowledge supplied by the recognition of named entities. Other works, such as (Iftene and Moruz, 2009) and our participation in the *Text Analysis Conference 2008* (Balahur et al., 2008), have also proven that knowledge about named entities positively helps in modeling entailments. In our case, rather than constructing the system based on named entity inferences, we study the addition of this knowledge in our textual entailment recognition system.

Therefore, similarly as we did for verbs, we explored ways to find out entity counterparts between the text and the hypothesis. The first step is to recognize named entities, and for this purpose we use our in-house named entity recognizer, called NERUA (Kozareva et al., 2007). Afterwards, we use two surface techniques to discover NE relations: partial entity matching and acronym correspondences between the NEs detected in the hypothesis and the ones in the text.

3.3 Corpus pre-processing

We have identified some scenarios where it would be beneficial to perform additional corpus preprocessing. These are described as follows.

Handling misspellings Given that our method is heavily based on term frequencies, a misspelling in the processed documents could introduce a high level of noise, since they will have a lower document frequency, and therefore a higher idf. Also, if a misspelling appears in a suspicious and a source document, these will be heavily linked by this term, and their similarity score may not be fair when comparing it with other documents. There-

fore, it would be beneficial to apply a spell corrector over the documents in our corpora, such as the one described in (Gao et al., 2010). To minimize the impact of false positives from the speller system, we would perform a two-pass algorithm. In the first pass we would not apply the spell corrector, and would try to retrieve all the plagiarisms that our system recognizes. In the second pass we would apply the spell corrector and attempt to extract additional appropriations. By doing this we ensure that we don't loose plagiarisms if the spell corrector system introduces some noise into the data.

Document translation When plagiarizing a document, an author can choose to translate it into a different language. This is the case, for instance, for some of the plagiarized documents of the PAN corpora, which have been translated into Spanish or German (Potthast et al., 2009). These appropriations won't be detected by our system unless we translate them into English, as this is the language in which the source documents are written. As a pre-processing step, we propose to apply a language detector over the set of suspicious documents, and if this tool detects that they are not in English, we execute an automatic translator to transform the corresponding document into English. The detection step is performed using the API of a machine translation application. Given that this is a remote live production system and some of the documents in our corpus can be large, sending the whole text doesn't seem to be the best approach. For the user case where we have a large amount of suspicious documents to process, we send a fragment composed of the first few hundreds of words from a document in order to get a fast and scalable response. This is not completely accurate, as some times documents contain fragments written in different languages. If we only process one suspicious document, we perform a more complex and accurate process. To do this we first split the document content into sentences based on punctuation symbols. Then, we submit three random sentences from the text to the translation application. If all of them return the same language detected, this will be the one of the document. If this is not the case we take another set of three sentences. Similar to what we previously mentioned, we perform a two-pass algorithm in order to reduce the impact of false positives introduced by the translation software.

4 Experimentation and results

As mentioned before, the corpora that we have used to measure and evaluate our system have been provided by the *1st International Competition on Plagiarism Detection*. These are composed thousands of source and suspicious documents, some of the latter containing automatically generated plagiarisms with different levels of obfuscation. In addition, some source documents are written in Spanish or German, but the corresponding plagiarized document has been translated into English.

4.1 Baseline system

To experiment with our system we used the external plagiarism corpora from the *1st International Competition on Plagiarism Detection*. The first aspect we experimented with was trying to determine the optimal number of documents to be selected, given that a larger amount would lead to higher accuracy, but would affect performance negatively. The opposite applies to smaller selected document sets.

Table 1 shows the results from this experiment using different set sizes, where column *Captured* represents the number of plagiarisms that are contained within the set of source documents, and *Missed* those that are not included in this set.

Size	Recall	Captured	Missed
1	0.3260	23,970	49,552
5	0.6875	50,547	22,975
10	0.7781	57,206	16,316
20	0.8282	60,893	12,629
30	0.8479	62,340	11,182
40	0.8595	63,189	10,333
50	0.8698	63,947	9,575
60	0.8760	64,403	9,119
70	0.8820	64,843	8,679
80	0.8869	65,205	8,317
90	0.8905	65,473	8,049
100	0.8941	65,734	7,788

Table 1: Metrics using different selected document set sizes.

Given the values from Table 1, we decided to use a number of documents of 10, since we believe it is the best trade-off between amount of texts and recall. After this step, we executed the passage detection, which produced an overall score of 0.3902. As we can see in these results, the

strongest aspect of our baseline system is its precision, where it ranks the third among all participants. On the other hand, recall and granularity were not as good, but still within the top half. The reason why recall is lower is in part due to the fact that we chose 10 source documents per suspicious text to evaluate, giving a maximum coverage value of 77.81%. Apart from this, and since our method is purely lexical, we miss plagiarisms that are not written in similar ways. Finally, documents that are translated will also lower our recall. On the other hand, granularity would have been lower if we had been more aggressive at merging matches, although then precision might have suffered.

4.2 Applying a textual entailment recognition system

Due to the expensive computational cost of executing a textual entailment recognition system, we used the corpora provided for the *Recognizing Textual Entailment* challenges. To simulate that the text-hypothesis pairs in these corpora are documents, we combine the texts into a single document and the hypothesis into another one, and then perform a plagiarism detection using both documents. Table 2 shows the results using our baseline system and the textual entailment recognition method previously described. As we can see in this table, our baseline system doesn't recognize the cases where there is an entailment, given that the pairs are written in a very different way. Applying our textual entailment recognition method provides significant gains.

Corpora	System	Accuracy
RTE-2	Baseline System	0.5000
	Textual Entailment	0.6125
RTE-3	Baseline System	0.5125
	Textual Entailment	0.6800
RTE-4	Baseline System	0.5000
	Textual Entailment	0.6250
RTE-5	Baseline System	0.5000
	Textual Entailment	0.6350

Table 2: Results of our baseline and textual entailment systems using the RTE test corpora.

4.3 Handling misspellings

Given the nature of the corpora provided for the *1st International Competition on Plagiarism Detection*, we cannot apply them to test a speller system given that the plagiarisms are automatically generated and therefore they do not contain misspellings (Potthast et al., 2009). Instead, we evaluate the addition of this module based on the results that spellers achieve in real-world applications.

Typically, web spellers have an accuracy of around 90% assuming an 85% of correctly spelled queries and 15% of misspellings, as described in (Gao et al., 2010). This means that there is clearly a gain of applying these systems as, even though they introduce some noise, in general terms they produce significant benefits. In addition, they are deterministic systems, and given that we apply them to both the source and suspicious document, an incorrect behavior for a given word in a source document would also be applied to the same word in the suspicious, and vice versa. In our system we want to match terms that appear in the same manner, and therefore a false positive or negative produced by the speller system won't hurt the accuracy of our plagiarism detection software.

Assuming a highly misspelled document, the application of a speller could produce a net gain of about 5%, which is a very important increase. In addition, speller systems typically return a normalize score value depending on the confidence of a given candidate. Based on this they either produce a suggestion, when there is lower confidence, or an auto-correction, when there is higher. We could tune our system to use a more or less aggressive speller depending on the user's needs as well as the nature of the input corpora.

4.4 Document translation

The corpora provided for the *1st International Competition on Plagiarism Detection* contains source documents in languages other than English, although the suspicious ones have been translated. Concretely, there are $13,559$ source documents in English, and 870 in other languages. Given that the suspicious texts will be in English, our system won't find the plagiarisms associated to those 870 due to language mismatches. To overcome this issue we applied the translator previously described, using different configurations. The parameter we changed was the number of words from the document to submit to the translator, using the first 200, 500 and $1,000$ words.

The following table shows the results from applying the language detector over the source documents corpus.

System	Accuracy	Correct	Incorrect	TP	TN	FP	FN
Baseline (no detection)	0.9397	13,559	870	0	13,559	0	870
Detection ($\|words\| = 200$)	0.9936	14,337	92	816	13,521	38	54
Detection ($\|words\| = 500$)	0.9967	14,381	48	843	13,538	21	27
Detection ($\|words\| = 1,000$)	0.9974	14,392	37	847	13,545	14	23

Table 3: Results from applying the language detector over the source documents corpus.

We define positives as the documents that have been translated, and negatives those that have been not. In this table we can see that there is a 5.77% increase in accuracy if we apply a language detector using the first $1,000$ words of a document. However, given that we use a two-pass algorithm, the number of FPs would be 0, which means that the final accuracy after applying a language detection software would be 0.9984, which is a 5.87% higher than the baseline. This means that, assuming a perfect translator and plagiarism detector, our system's score could increase in almost six points, which is a big improvement. The final gain will depend on the user's document translation software choice.

5 Conclusions and future work

In this paper we have presented a baseline system for external plagiarism analysis mainly based on lexical similarities, and a set of more advanced techniques that could be beneficial to external plagiarism analysis. While the baseline system is very efficient and produces reasonable results, the application of the aforementioned advanced techniques can have a very significant impact, depending on the corpus' nature. However, these latter methods decrease our overall system's performance considerably, so they are not applicable to large corpora.

We have also explained two scenarios where we believe that plagiarism detection tools are applied. In the first of them, where we would have a large suspicious documents corpus, the application of advanced techniques would not be feasible given their low efficiency. Therefore, in this case we would have to use our baseline system which is mainly based on lexical measures. On the other hand, in the second user scenario, where we only have one suspicious document to analyze, the application of the aforementioned advanced techniques is suitable given the smaller amount of data to process. In this case we will be able to achieve higher accuracy rates and support a larger number of obfuscation cases. Therefore, there is a trade-off between accuracy and response time, which will be in large determined by the size of the corpus to process.

As future work we would like to apply a word alignment algorithm to detect plagiarisms, such as the one described in (Och, 2002). This would be a more flexible and accurate approach, rather than forcing the words to appear in the same position in both documents being analyzed, although its computational cost would also be considerably higher. This should allow our system to recognize higher levels of obfuscation than our current approach. In addition, it would be very beneficial for multilingual plagiarism analysis. This kind of task presents the challenge that words might not appear in the same order, not even after a machine translation tool has been applied. Hence, applying the aforementioned word alignment algorithm would allow us to handle better multilingual plagiarism.

Acknowledgements

This research has been partially funded by the Spanish Ministry of Science and Innovation (grant TIN2009-13391-C04-01) and the Conselleria d'Educació of the Spanish Generalitat Valenciana (grants PROMETEO/2009/119 and ACOMP/2010/286). Furthermore, we would like to thank Dario Bigongiari and Michael Schueppert for their help and support.

References

Alexandra Balahur, Elena Lloret, Óscar Ferrández, Andrés Montoyo, Manuel Palomar, and Rafael Muñoz. 2008. The DLSIUAES Team's Participation in the TAC 2008 Tracks. In *Notebook Papers of the Text Analysis Conference, TAC 2008 Workshop*, Gaithersburg, Maryland, USA, November.

Chiara Basile, Dario Benedetto, Emanuele Caglioti, and Mirko Degli Esposti. 2008. An example of mathematical authorship attribution. *Journal of Mathematical Physics*, 49:125211–125230.

Chiara Basile, Dario Benedetto, Emanuele Caglioti, Giampaolo Cristadoro, and Mirko Degli Esposti. 2009. A Plagiarism Detection Procedure in Three Steps: Selection, Matches and "Squares". In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 19–23, San Sebastián (Donostia), Spain, September.

Courtney Corley and Rada Mihalcea. 2005. Measuring the Semantic Similarity of Texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18, Ann Arbor, Michigan, USA, June.

Óscar Ferrández, Daniel Micol, Rafael Muñoz, and Manuel Palomar. 2007a. A Perspective-Based Approach for Solving Textual Entailment Recognition. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 66–71, Prague, Czech Republic, June.

Óscar Ferrández, Daniel Micol, Rafael Muñoz, and Manuel Palomar. 2007b. DLSITE-1: Lexical Analysis for Solving Textual Entailment Recognition. In *Natural Language Processing and Information Systems*, volume 4592, pages 284–294.

Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirck, and Xu Sun. 2010. A Large Scale Ranker-Based System for Search Query Spelling Correction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 358–366, Beijing, China, August.

Cristian Grozea, Christian Gehl, and Marius Popescu. 2009. ENCOPLOT: Pairwise Sequence Matching in Linear Time Applied to Plagiarism Detection. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 10–18, San Sebastián (Donostia), Spain, September.

Andrew Hickl and Jeremy Bensley. 2007. A Discourse Commitment-Based Framework for Recognizing Textual Entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171–176, Prague, Czech Republic, June.

Adrian Iftene and Mihai-Alex Moruz. 2009. UAIC Participation at RTE5. In *Notebook Papers of the Text Analysis Conference, TAC 2009 Workshop*, Gaithersburg, Maryland, USA, November.

Jan Kasprzak, Michal Brandejs, and Miroslav Křipač. 2009. Finding Plagiarism by Evaluating Document Similarities. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 24–28, San Sebastián (Donostia), Spain, September.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. Extending Verbnet with Novel Verb Classes. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, June.

Z. Kozareva, Ó. Ferrández, A. Montoyo, and R. Muñoz. 2007. Combining data-driven systems for improving Named Entity Recognition. *Data and Knowledge Engineering*, 61(3):449–466.

Daniel Micol, Óscar Ferrández, and Rafael Muñoz. 2007. DLSITE-2: Semantic Similarity Based on Syntactic Dependency Trees Applied to Textual Entailment. In *Proceedings of the TextGraphs-2 Workshop*, pages 73–80, Rochester, New York, USA, April.

Daniel Micol, Óscar Ferrández, Fernando Llopis, and Rafael Muñoz. 2010. A Lexical Similarity Approach for Efficient and Scalable External Plagiarism Detection. In *Proceedings of the SEPLN'10 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, Padua, Italy, September.

Markus Muhr, Mario Zechner, Roman Kern, and Michael Granitzer. 2009. External and Intrinsic Plagiarism Detection Using Vector Space Models. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 47–55, San Sebastián (Donostia), Spain, September.

Franz Josef Och. 2002. *Statistical machine translation: from single-word models to alignment templates*. Ph.D. thesis, RWTH Aachen.

Martin Potthast, Benno Stein, Andreas Eiselt, Alberto Barrón Cedeño, and Paolo Rosso. 2009. Overview of the 1st international competition on plagiarism detection. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 1–9, San Sebastián (Donostia), Spain, September.

Martin Potthast, Benno Stein, Andreas Eiselt, Alberto Barrón Cedeño, and Paolo Rosso. 2010. Overview of the 2nd international competition on plagiarism detection. In *Proceedings of the SEPLN'10 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, Padua, Italy, September.

Álvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. 2008. Towards an Entity-based recognition of Textual Entailment. In *Notebook Papers of the Text Analysis Conference, TAC 2008 Workshop*, Gaithersburg, Maryland, USA, November.

Efstathios Stamatatos. 2009. Intrinsic Plagiarism Detection Using Character n-gram Profiles. In *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse*, pages 36–37, San Sebastián (Donostia), Spain, September.