CoReMo System
(Contextual Reference Monotony)
Lab Report for PAN at CLEF 2010

Diego Antonio Rodríguez Torrejón * º, José Manuel Martín Ramos º

* I.E.S. "José Caballero", º Universidad de Huelva (Spain)
diego@dartsystems.es jmmartin@dti.uhu.es

Abstract. In this paper a new approach is shown for a very fast monolingual external plagiarism detection system based on an altered n-gram concept (contextual n-gram), a new high precision contextual Information Retrieval engine, and a new pruning strategy (Referential Monotony) for plagiarism detection and its limits. The assessment results can be compared with the carried out by the winner team at PAN'09, but achieved with remarkable speed (35 min) and low hardware requirements (single laptop).

Keywords: plagiarism detection, n-gram, contextual n-gram, Referential Monotony, Information Retrieval

1 Introduction

In this paper, a new only external and monolingual plagiarism detection system is shown. Its goal is minimizing hardware resources and however, getting fastly high PAN performance measures. It's based on three innovative proposals:

1. ***Contextual n-gram*:** a new n-gram concept modification with two main features: describes the sentence context where it's on, and it is a highly discriminative fingerprint for its sentence between the others into a very wide collection.
2. **New Information Retrieval System** (IRS) with very high precision, specific for this goal, based on former concept.
3. ***Referential Monotony*** (RM): new pruning strategy to find plagiarism limits.

The performances are in general better than PAN'09 best ranked teams, but got with a much lower computational cost. Because that, it's an interesting opportunity.

2 External Plagiarism Detection

In [1], the generic process for external plagiarism analysis is presented. The system shown in this paper is well fitting that schema.

2.1 "Contextual N-gram" for Designing an Information Retrieval System.

The system shown in this paper, bases the plagiarism analysis on n-grams comparison [2], but by using an special treatment to build them.

The "contextual n-gram" denomination is referred to the feature of these n-grams to describe the essential context with a very short group of word stems.

Because making n-grams by simple tokens extraction, gets analyzers very vulnerable to obfuscation, six steps are carried out when modeling documents by n-grams in order to improve the essential context definition, with gets highly useful n-grams to locate possible plagiarism, obfuscated or not:

1. **Lowercase folding** when tokenization (a very common practice).
2. **Empty words (*stopwords*) filtering.** This step gets n-grams much more context definitory. *Stopwords* are also easy to delete/change for obfuscation purposes.
3. **One character tokens filtering.** As they are very used in enumerations and having high frequency, they get few contextual meaningful.
4. **Stem reduction** (*stemming)* [3] contributes to get better recall for detecting plagiarisms when words are changed by derivative ones.
5. **Alphabetic tokens order into every n-gram**, processing the result as canonical representative for the all possible tokens permutations set. This step reduces effectivity of words order changes due to sentence rewriting or translation [4], improving recall.
6. ***n – 1* tokens overlapping** (on its natural order) to extract consecutive contextual n-grams, gets better detection on former obfuscation types.

May be thought that all these steps to help improving recall, may also get false positive increasing, but it is experimentally demonstrated that steps 2 and 3 gets enough compensation due to the final discriminative capacity got by contextual n-gram.

Studying PAN'09 corpuses, it was probated that a high percentage of this n-gram type behaves like a fingerprint for the passage/document they belong, specially if n-grams are 3th grade or higher. (figure 1).

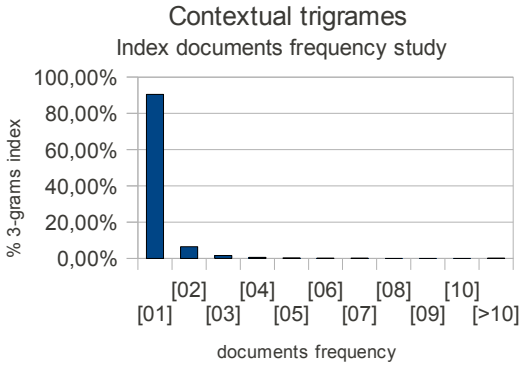

Figure 1: PAN'09 Development corpus Contextual trigrams index

This discriminative capacity is strengthened by the neighbor contextual n-grams, being an excellent base to develop specific IRS to detect and locate plagiarisms.

By analyzing the PAN'09 development corpus, it was found that the probability for to repeat a contextual trigram from a new non plagiarized document in a concrete existing document from corpus, was of 0,0026%. However, if it's a plagiarized one,

the probability to point to the correct source document is 94,37% (using only three document references max. for every trigram). Contextual bigrams gets 0,052% to be repeated in a concrete document, and 77,00% or 77,76% for pointing to correct document when existing plagiarism (using 5 or 9 references max respectively).

Contextual n-gram groups are excellent for high precision calculation of the most similar passage/document by contextual similarity.

The only inconvenience is the necessary index size, several times bigger than the own text corpus. However, an approximation to VSM, but only based on *df*[1] proportional weighting, and a limited document references amount, is enough to get a very high precision IRS with lower memory consumption (index stores *df* and 2~3 max. source references if using trigrams or 5~9 if bigrams).

As plagiarizers used to take several plagiarism fonts, having an available IRS as proposed in this paper, to reduce and strength the search space [5], a change of strategy is preferable instead of returning a fix number of candidate fonts: After splitting the suspicious document, identifying only one candidate as possible source (for every split) and using the RM pruning strategy.

2.2 Referential Monotony (RM)

When analyzing, it is highly probable that a big number of suspicious fragments should have a possible source document associated. Analyzing anything would need a lot of time and/or computational resources, getting many false positives. To avoid this annoyance, a new pruning strategy is used, named Referential Monotony, consisting in rejecting (as casual matching) all suspicious fragments appearing alone, without repeating reference at least certain times (monotony threshold). RM gets a fast filtering of so enough wide suspicious sections as to point that there is a continuous high correspondence with the right source document, getting also a fast gross detection for its limits.

In the figure 2, the detection process and search space reduction by RM is shown: dark gray emphasized splits give direct detection (5 consecutive splits pointing to reference doc #91) due to pass RM threshold (4 in this example). Light gray splits are included for fishing possible words out of direct detection borders.

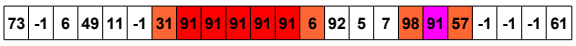

Fig. 2: Only source candidate per split (basic for RM) - Recall improved by *feedback*

The system presented excellent results by using only this strategy with a split length of 25 contextual n-grams and 4 times for RM threshold. The annoyance for this strategy is that plagiarized fragments shorter than 4 splits are not detected (75 contextual n-grams, or about 150 words).

To detect short length (but almost verbatim) plagiarisms, RM threshold is reduced for a zone when any split gets high similarity value.

As many plagiarizers use to employ same fonts to take several fragments, after getting sources knowledge by RM from greater zones, *feedback* for a second pass may be arranged for fishing the smaller, as show in figure 2 left zone.

---

1 Frequency of a term based in the number do documents containing it.

Our preliminary version for this idea (a necessary filtering is not yet implemented) only gets similar overall results (+/- 0,5%), but with a better precision/recall balance.

2.3 Time Processing Reduction

Although RM prune is the main reason, this goal was improved by several strategies:

- Using **C (gcc) 64 bits** version on **GNU - Linux** with **ext4 filesystem**.
- Using an **unbalanced binary tree** to sort n-grams and building partial inverted index, later mixed as **ordered vector**s for **final inverted index**.
- **IRS uses binary search** on former vector to locate n-grams matching with source documents. This gets similar efficiency order that a balanced tree but using less memory.
- Avoid repeating source-documents n-grams conversion by saving n-gram versions (while indexing), and caching last 50 analyzed.

Hardware and software used for development and PAN2010 competition:

- Acer Aspire laptop 5920G (Intel T5750 2.0 GHz processor, 4GB RAM), upgraded to 7200 rpm 2,5" HD.
- Ubuntu GNU-Linux 10.04 64 bits edition using EXT4 file-systems.
- GNU – C language. Netbeans 6.8 as IDE and Valgrind were used.

2.4 Plagiarism Detection Process

External analysis process is arranged by these main steps:

1. Language documents classification (to discard non English sources).
2. Building monolingual inverted index, saving on disk and memory loading.
3. Splitting suspicious documents by fix amount of contextual n-grams.
4. Using the new IRS to get only one source document for every split.
5. Determination of plagiarism existence by Referential Monotony.
6. Finding plagiarism border n-grams separately for suspicious zone by double search from start and end of detected gross zone over the IRS.
7. Using suspicious detected zone to search the best matching window into source document, getting better recall and precision than in suspicious section. Then a post-refinement is done for suspicious fragment limits.
8. Saving analysis results in XML files.
9. Evaluate results over training XML gold standard (if available).

3 Evaluation

Thanks to its speed, more than 150 trials were carried out on PAN'09 training corpus while system development (started last year), with contextual n-grams grade 2 and 3.

The best PAN performance is got by contextual trigrams, however time and resources are better by bigrams. Ten folds[2] analysis confirmed behavior regularity.

PAN-PC-09 was only used to confirm former tweaks and prospects, on a 8GB RAM PC. For the competition, we used once again the same 4GB RAM laptop.

Table 1 shows results got in several corpuses and hardware used summary.

As when writing this paper, no information is available for costs, hardware and times got by other PAN2010 teams, figure 3 shows this for best ranked teams in PAN'09, compared to this system by trigrams (T) or bigrams (B). This comparative was estimated [6] by the available information from [7], [8] and [9].

No cross-lingual performance is expected. As experimental version gets lower *plagdet score* than monolingual, we used the last one (where non English source documents are excluded).

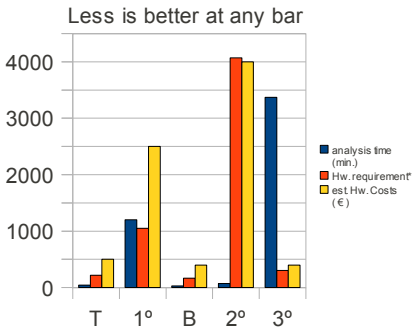

Fig.3: PAN'09 comparative Analysis time – Hardware reqs. – Costs

Overall performance got at **PAN2010: 0.5851** *(non official external: 0.6666).*

Timing: *Indexing (1.6 GB) 20 min 06 sec + Analysis (3.2 GB) 55 min 44 sec*

4 Conclusions

Good results, with remarkable high speed and low resources. Waiting improvements:

- Feedback filtering (+ recall and overall).
- Multilingual version refinement (+ recall and overall).
- Concurrent and/or parallel programing for multi-core machines (+ speed).
- Using SSD (Solid State Disk) HD (+ speed).

Including a *confidence attribute* in detections would help to users and enforces hybrid analyzers development without present penalty.

Contextual n-grams could improve other NLP disciplines as clustering, classifying, reply oriented search, etc. RM prune should be also usefull in other fields.

5 Acknowledgements

Authors thanks to PAN-PC corpuses development team (excellent resources) and the rest of PAN organization and competitors for their motivative work and papers.

2 Similar size subsets got by dividing the suspicious corpus.

Table 1: detailed features for analyzing different corpuses, by different computers using contextual n-grams grade 2 and 3

PAN corpus	**'09 develop.**	**'09 develop.**	**'09 compet.**	**'09 compet.**	**'09 compet.**	**PAN-PC-09**	**'2010 external***	**'2010 global**
Source files	7214 (1.1 GB)	7214 (1.1 GB)	7214 (1.2 GB)	7214 (1.2 GB)	7214 (1.2 GB)	14429 (2.3 GB)	11148 (1.6 GB)	11148 (1.6 GB)
Suspicious files	7214 (1.5 GB)	7214 (1.5 GB)	7214 (1.4 GB)	7214 (1.4 GB)	7214 (1.4 GB)	14428 (2.9 GB)	15925 (3.2 GB)	15925 (3.2 GB)
Processor speed	2.0 GHz	2.0 GHz	2.0 GHz	**3.0 GHz**	2.0 GHz	**3.0 GHz**	2.0 GHz	2.0 GHz
RAM	4 GB	2 GB	4 GB	**8 GB**	2 GB	**8 GB**	4 GB	4 GB
parameters	n 3 l 25 m 4 F 1	n 2 l 30 m 4 F 1	n 3 l 25 m 4 F 1	n 3 l 25 m 4 F 1	n 2 l 30 m 4 F 1	n 3 l 25 m 4 F 1	n 3 l 25 m 4 F 1	n 3 l 25 m 4 F 1
Lang detection time	00min 37sec	00min 37sec	1min 24sec	0min 36sec	1min 24sec	1min 09sec	2min 17sec	2min 17sec
Indexing time	13min 02sec	10min 31sec	13min 18sec	09min 47sec	10min 09sec	21min 00sec	20min 06sec	20min 06sec
Analysis time	18min 01sec	16min 10sec	18min 22sec	12min 50sec	16min 19sec	29min 23sec	55min 44sec	55min 44sec
Total time	**31min 40sec**	**27min 18sec**	**33min 04sec**	**23min 13sec**	**27min 52sec**	**52min 32sec**	**78min 07sec**	**78min 07sec**
precision	0.7751	0.7067	0.7901	0.7901	0.7337	0.7152	0.8476	0.8507
recall	0.5685	0.5420	0.6497	0.6497	0.6231	0.5926	0.5505	0.4481
F-measure	0.6560	0.6123	0.7130	0.7130	0.6739	0.6481	0.6675	0.5870
granularity	1.0177	1.0172	1.0197	1.0197	1.0413	1.0286	1.0018	1.0044
overall	**0.6477**	**0.6060**	**0.7031**	**0.7031**	**0.6546**	**0.6351**	**0.6666**	**0.5851**
monolingual recall	0.6511	0.6207	0.6813	0.6813	0.6533	0.6489	n.a.	n.a.
Monolingual overall	0.6988	0.6528	0.7215	0.7215	0.6714	0.6668	n.a.	n.a.

n ngram grade – ***l*** split length – ***m*** RM threshold – ***F*** feedback activation – n.a. → not avaliable - * non official result

6 References

1. Potthast M., Stein A., Eiselt A., Barrón-Cedeño A., Rosso P.: Overview of the 1st International Competition on Plagiarism Detection. In: Stein B., Rosso P., Stamatatos E., Koppel M., and Agirre E. (eds.) *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09), pp. 1-9, Donostia-San Sebastian, Spain, September 2009*. CEUR-WS.org. ISSN 163-0073.(2009)
2. Barrón-Cedeño A., Rosso P.: On Automatic Plagiarism Detection based on n-grams Comparison. *Proc. European Conference on Information Retrieval, ECIR-2009,Springer-Verlag, LNCS* (5478) páginas 696-700. (2009)
3. Martin F. Porter. An algorithm for suffix stripping.(Porter stemmer) *Program*, 14(3):130-137. (1980) http://tartarus.org/~martin/PorterStemmer/index.html
4. Barrón-Cedeño A., Rosso P.: Monolingual and Crosslingual Plagiarism Detection - Towards the Competition. *Proc. III Jornadas PLN-TIMM, Madrid, Spain, February 5-6*, pages 29-32. (2009)
5. Barrón-Cedeño, A., Rosso P.: On the Relevance of Search Space Reduction in Automatic Plagiarism Detection. *Procesamiento del Lenguaje Natural*, 43:141-149. (2009)
6. Rodríguez-Torrejón D.A.: Detección de plagio en documentos. Propuesta de sistema externo monolingüe de altas prestaciones basada en n-gramas. Master Dissertation – Universidad de Huelva (2009)
7. Grozea, C., Gehl C., Popescu M.N.: ENCOPLOT pairwise sequence matching linear time plagiarism detection. In: Stein B., Rosso P., Stamatatos E., Koppel M., and Agirre E. (eds.) SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09), pp. 1-9, Donostia-San Sebastian, Spain, September 2009. CEUR-WS.org. ISSN 163-0073.(2009)
8. Kasprzak J., Brandejs M., Kripac M.: "Finding Plagiarism by Evaluating Document Similarities " (PAN'09 papers). In: Stein B., Rosso P., Stamatatos E., Koppel M., and Agirre E. (eds.) *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09), pp. 1-9, Donostia-San Sebastian, Spain, September 2009*. CEUR-WS.org. ISSN 163-0073.(2009)
9. Basile C, Benedetto D., Caglioti E.: A plagiarism detection procedure in three steps: selection, matches and 'squares'. In: Stein B., Rosso P., Stamatatos E., Koppel M., and Agirre E. (eds.) *SEPLN 2009 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09), pp. 1-9, Donostia-San Sebastian, Spain, September 2009*. CEUR-WS.org. ISSN 163-0073.(2009)