Overview of the 4th International Competition on Plagiarism Detection

Martin Potthast
Tim Gollub
Matthias Hagen
Jan Graßegger
Johannes Kiesel
Maximilian Michel
Arnd Oberländer
Martin Tippmann
Benno Stein

Webis Group
Bauhaus-Universität Weimar
www.webis.de

Parth Gupta
Paolo Rosso

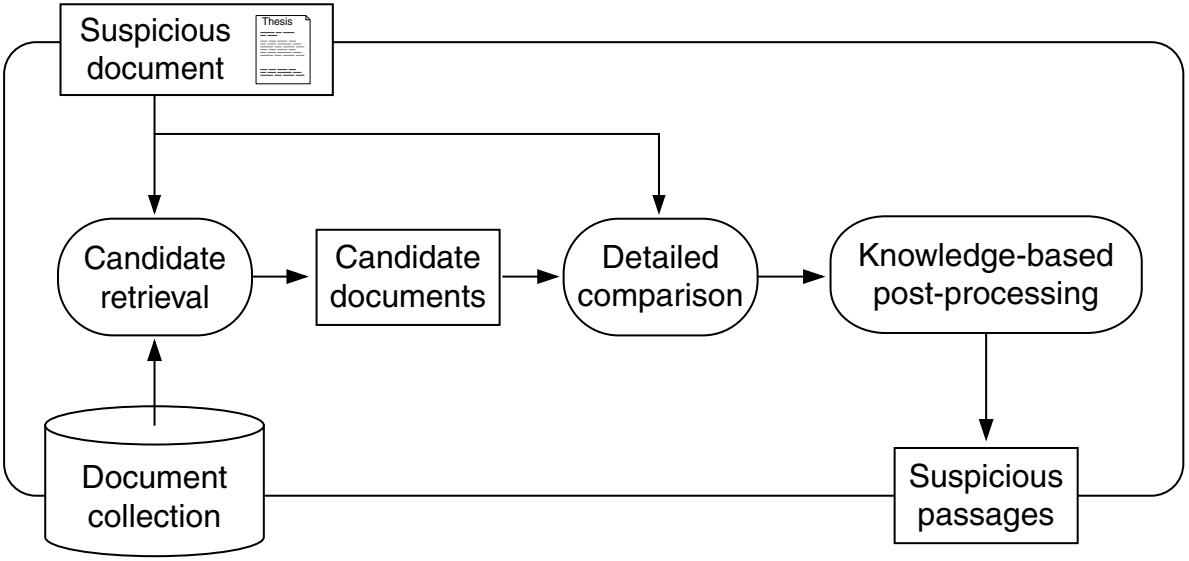
NLEL Group
Universitat Politècnica de València
www.dsic.upv.es/grupos/nle

Alberto Barrón-Cedeño

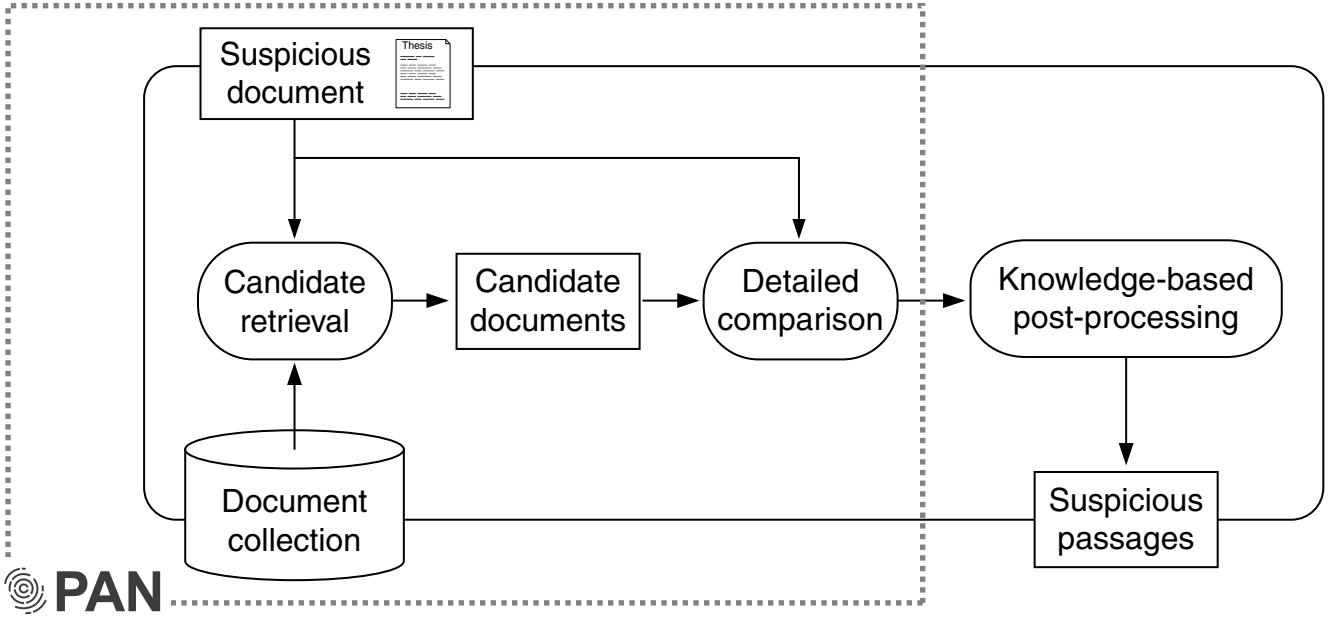
LSI Group
Universitat Politècnica de Catalunya
www.lsi.upc.edu

Introduction

Introduction



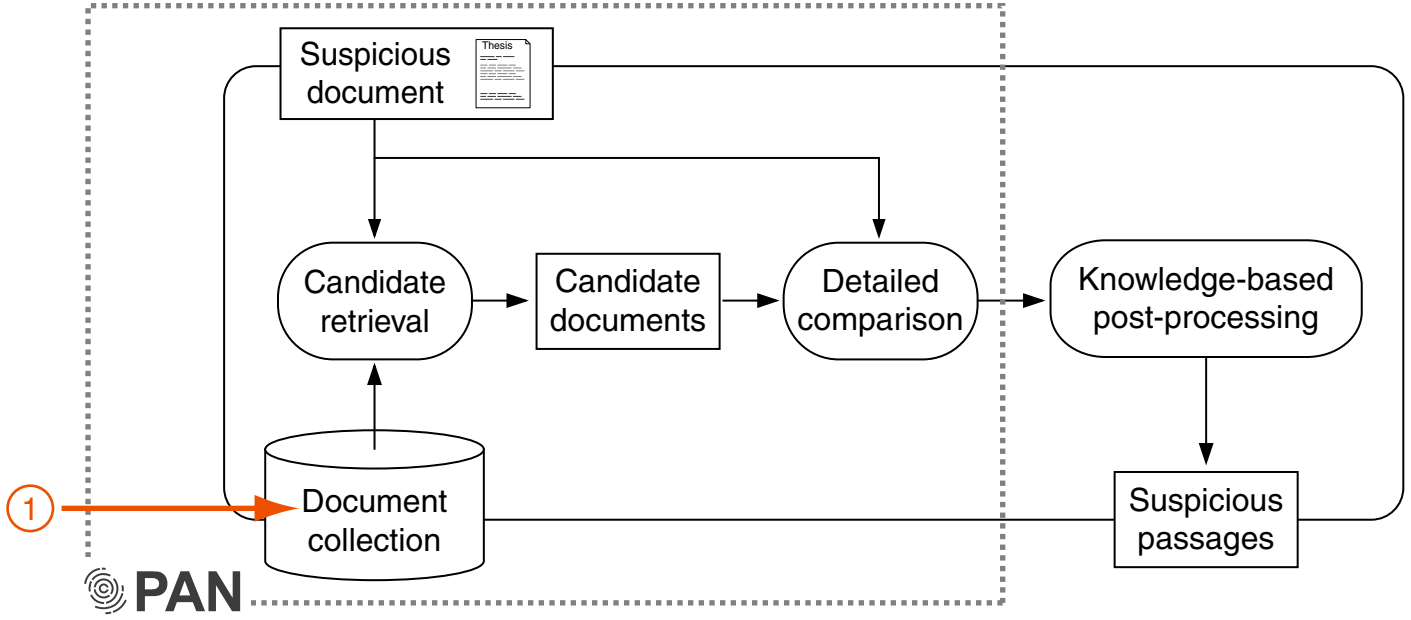
Introduction



Observations, problems:

1. Representativeness: the corpus consists of books, many of which are very old, whereas today the web is the predominant source for plagiarists.
2. Scale: the corpus is too small to enforce a true candidate retrieval situation; most participants did a complete detailed comparison on all $O(n^2)$ document pairs.
3. Realism: plagiarized passages consider not the surrounding document, paraphrasing mostly done by machines, the Web is not used as source.
4. Comparability: evaluation frameworks must be developed, too, and ours kept changing over the years, rendering the obtained results incomparable across years.

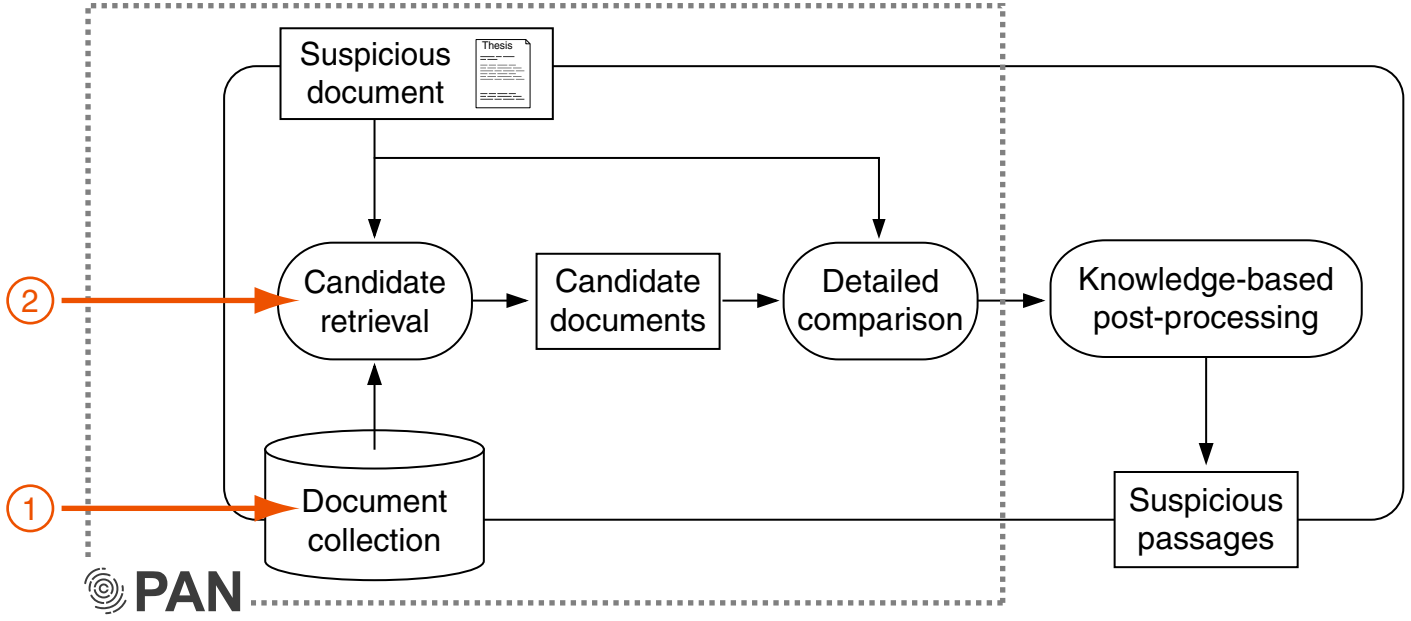
Introduction



Observations, problems:

1. Representativeness: the corpus consists of books, many of which are very old, whereas today the web is the predominant source for plagiarists.
2. Scale: the corpus is too small to enforce a true candidate retrieval situation; most participants did a complete detailed comparison on all $O(n^2)$ document pairs.
3. Realism: plagiarized passages consider not the surrounding document, paraphrasing mostly done by machines, the Web is not used as source.
4. Comparability: evaluation frameworks must be developed, too, and ours kept changing over the years, rendering the obtained results incomparable across years.

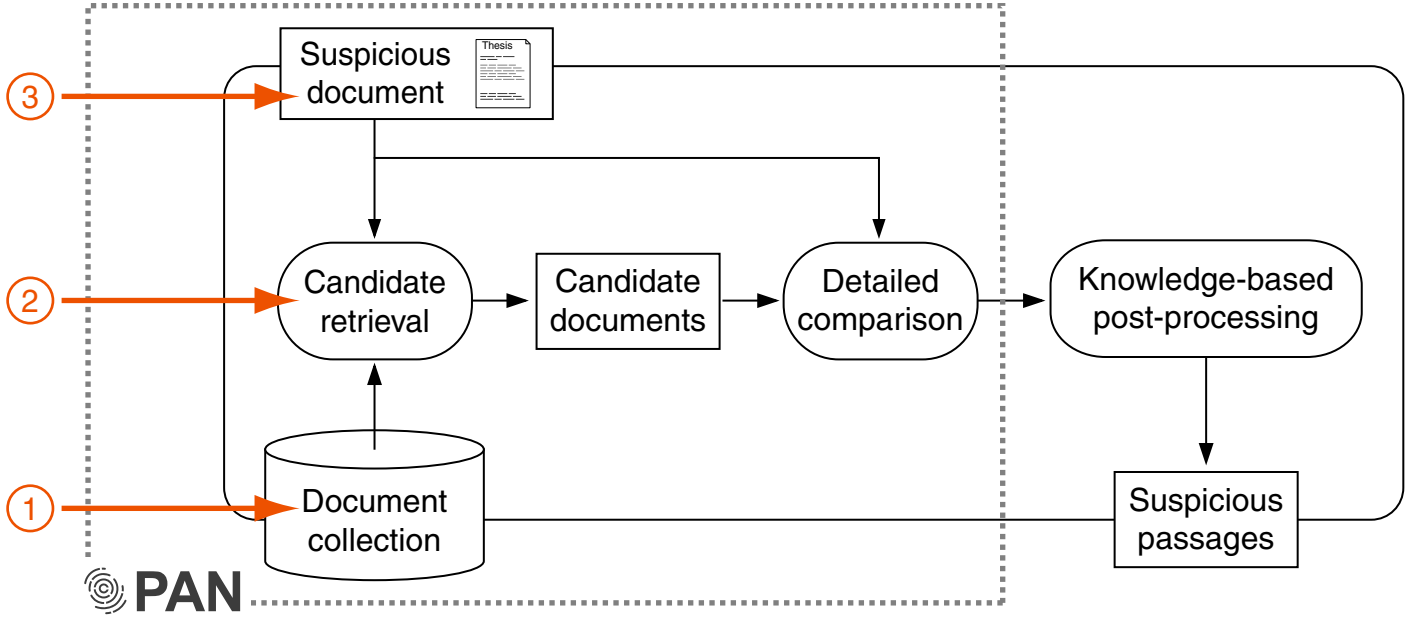
Introduction



Observations, problems:

1. Representativeness: the corpus consists of books, many of which are very old, whereas today the web is the predominant source for plagiarists.
2. Scale: the corpus is too small to enforce a true candidate retrieval situation; most participants did a complete detailed comparison on all $O(n^2)$ document pairs.
3. Realism: plagiarized passages consider not the surrounding document, paraphrasing mostly done by machines, the Web is not used as source.
4. Comparability: evaluation frameworks must be developed, too, and ours kept changing over the years, rendering the obtained results incomparable across years.

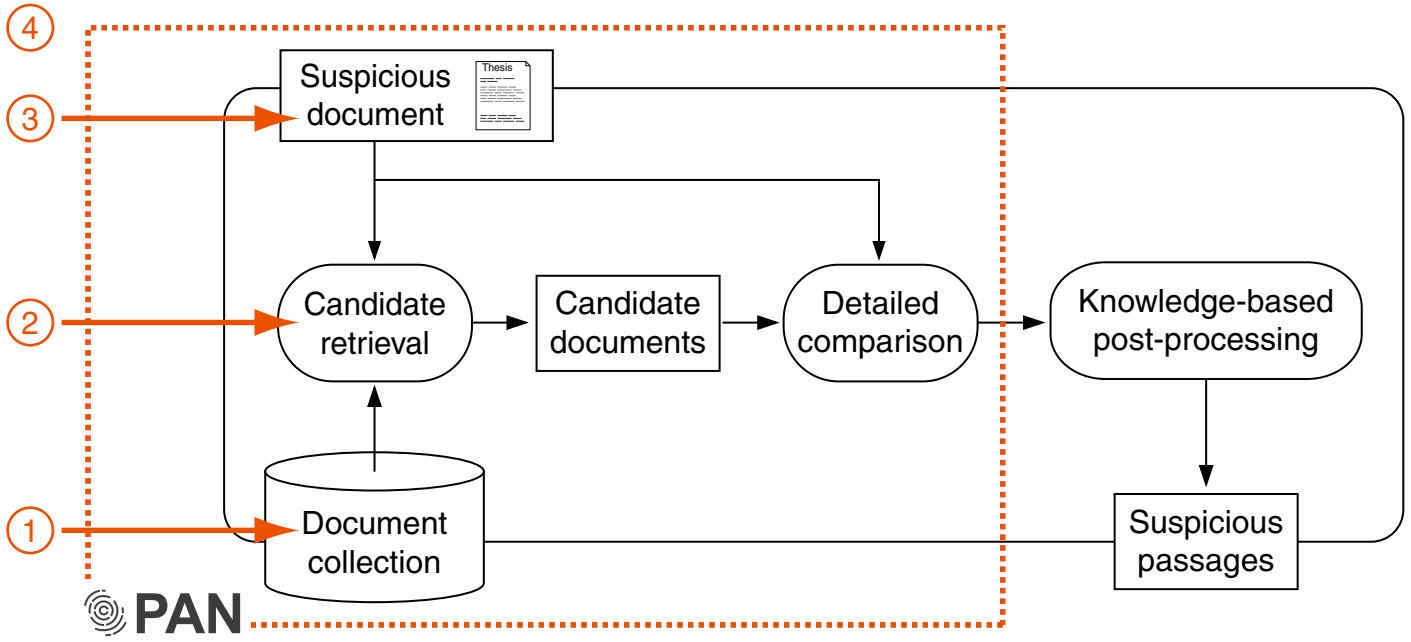
Introduction



Observations, problems:

1. Representativeness: the corpus consists of books, many of which are very old, whereas today the web is the predominant source for plagiarists.
2. Scale: the corpus is too small to enforce a true candidate retrieval situation; most participants did a complete detailed comparison on all $O(n^2)$ document pairs.
3. Realism: plagiarized passages consider not the surrounding document, paraphrasing mostly done by machines, the Web is not used as source.
4. Comparability: evaluation frameworks must be developed, too, and ours kept changing over the years, rendering the obtained results incomparable across years.

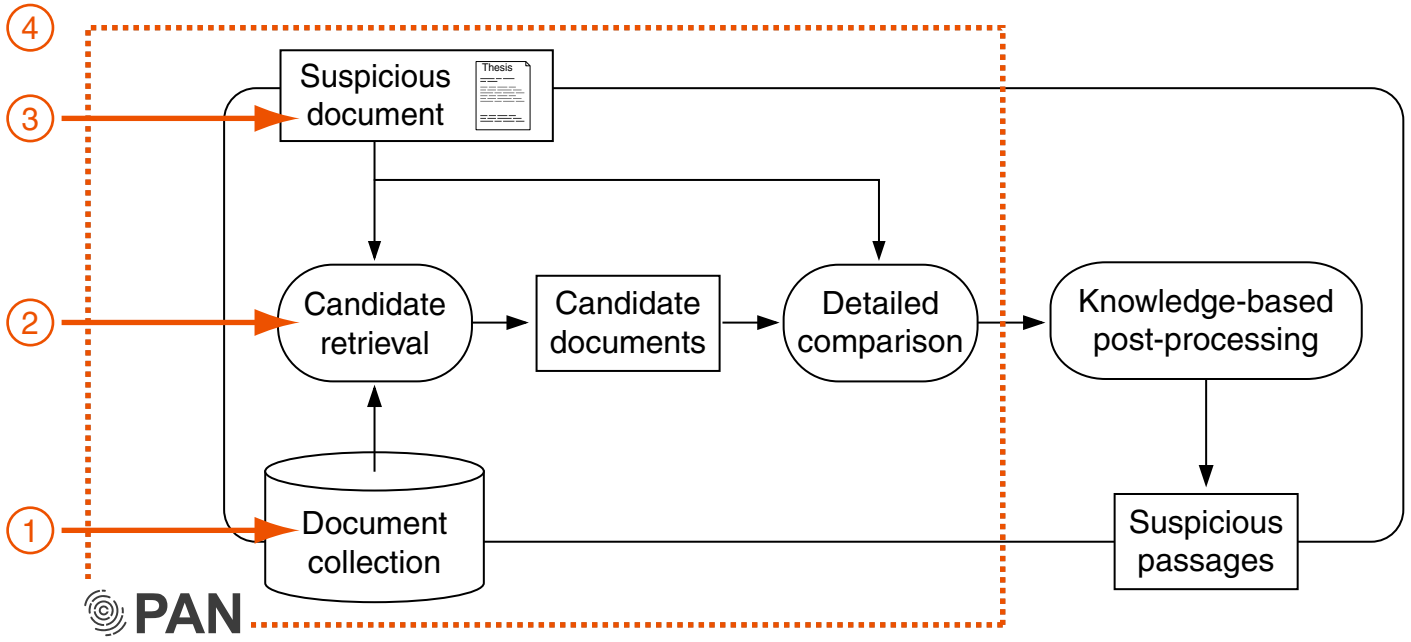
Introduction



Observations, problems:

1. Representativeness: the corpus consists of books, many of which are very old, whereas today the web is the predominant source for plagiarists.
2. Scale: the corpus is too small to enforce a true candidate retrieval situation; most participants did a complete detailed comparison on all $O(n^2)$ document pairs.
3. Realism: plagiarized passages consider not the surrounding document, paraphrasing mostly done by machines, the Web is not used as source.
4. Comparability: evaluation frameworks must be developed, too, and ours kept changing over the years, rendering the obtained results incomparable across years.

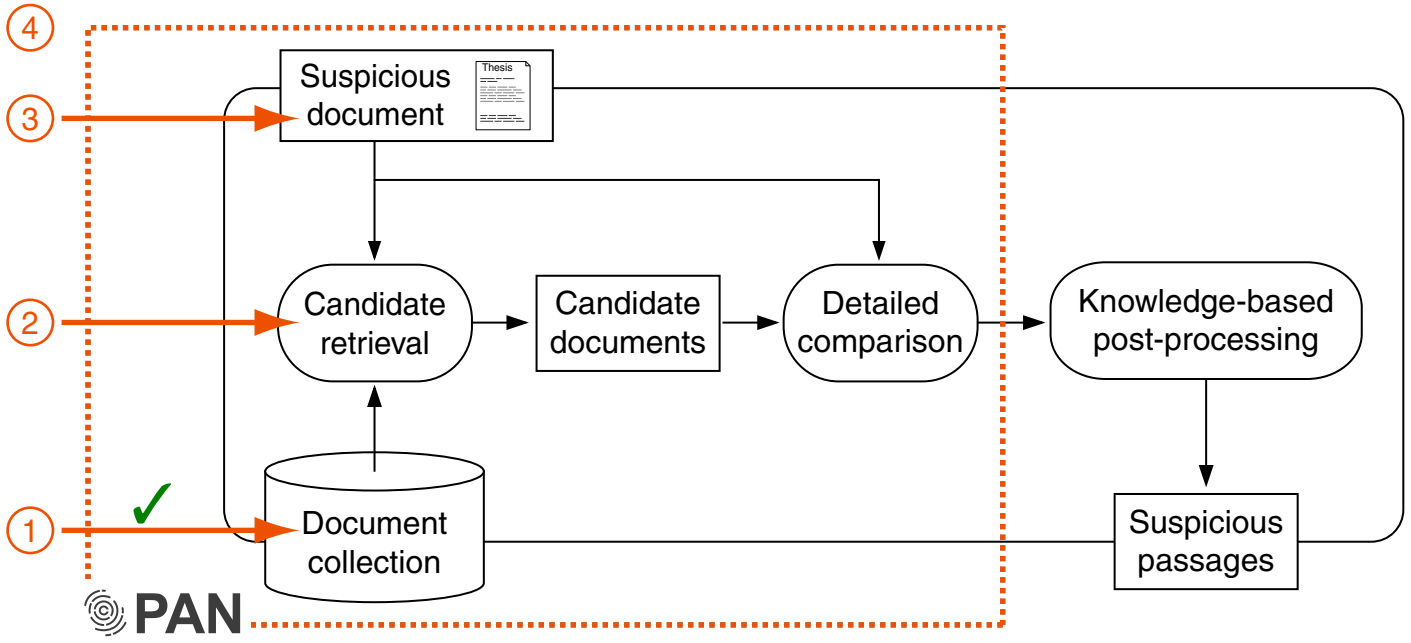
Candidate Retrieval



Considerations:

1. PAN'12 employed the English part of the ClueWeb09 corpus (used in TREC 2009-11 for several tracks) as a static Web snapshot. Size: 500 million web pages, 12.5TB
2. Participants was given efficient corpus access via the API of the ChatNoir search engine. ClueWeb and ChatNoir ensured experiment reproducibility and controllability.
3. The new corpus: manually written digestible texts, topically matching plagiarism cases, Web as source (for document synthesis and plagiarism detection).

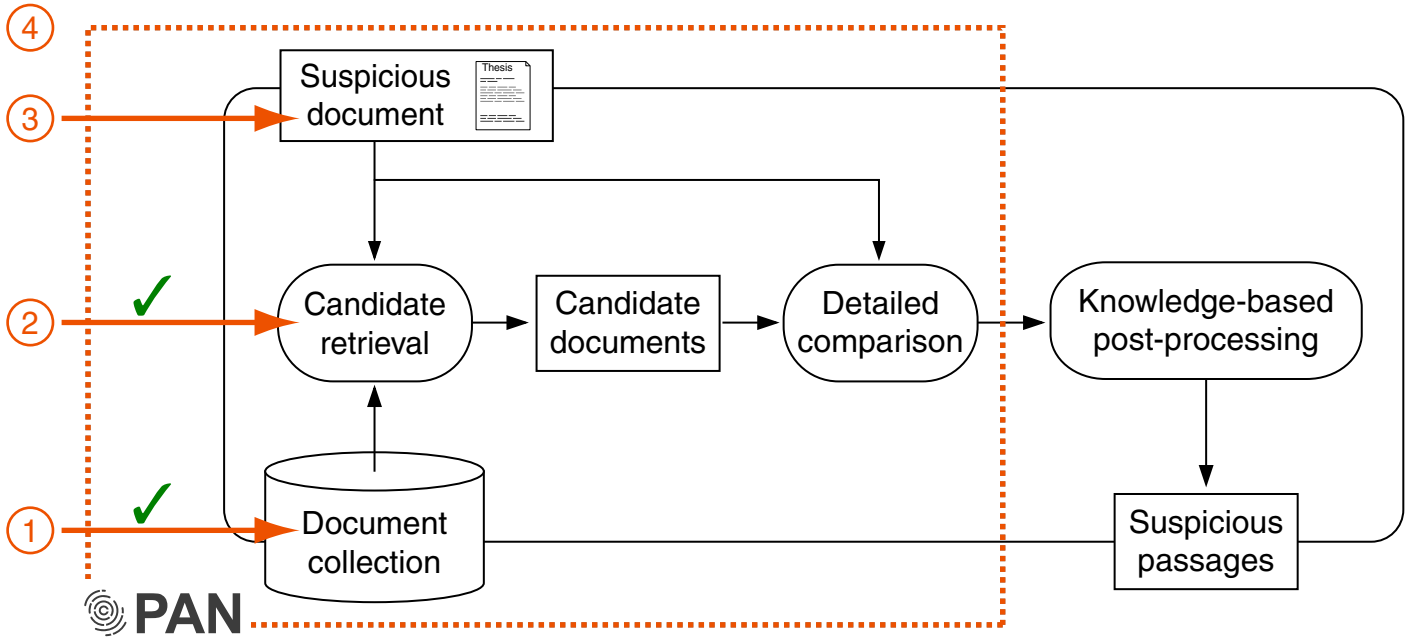
Candidate Retrieval



Considerations:

1. PAN'12 employed the English part of the ClueWeb09 corpus (used in TREC 2009-11 for several tracks) as a static Web snapshot. Size: 500 million web pages, 12.5TB
2. Participants was given efficient corpus access via the API of the ChatNoir search engine. ClueWeb and ChatNoir ensured experiment reproducibility and controllability.
3. The new corpus: manually written digestible texts, topically matching plagiarism cases, Web as source (for document synthesis and plagiarism detection).

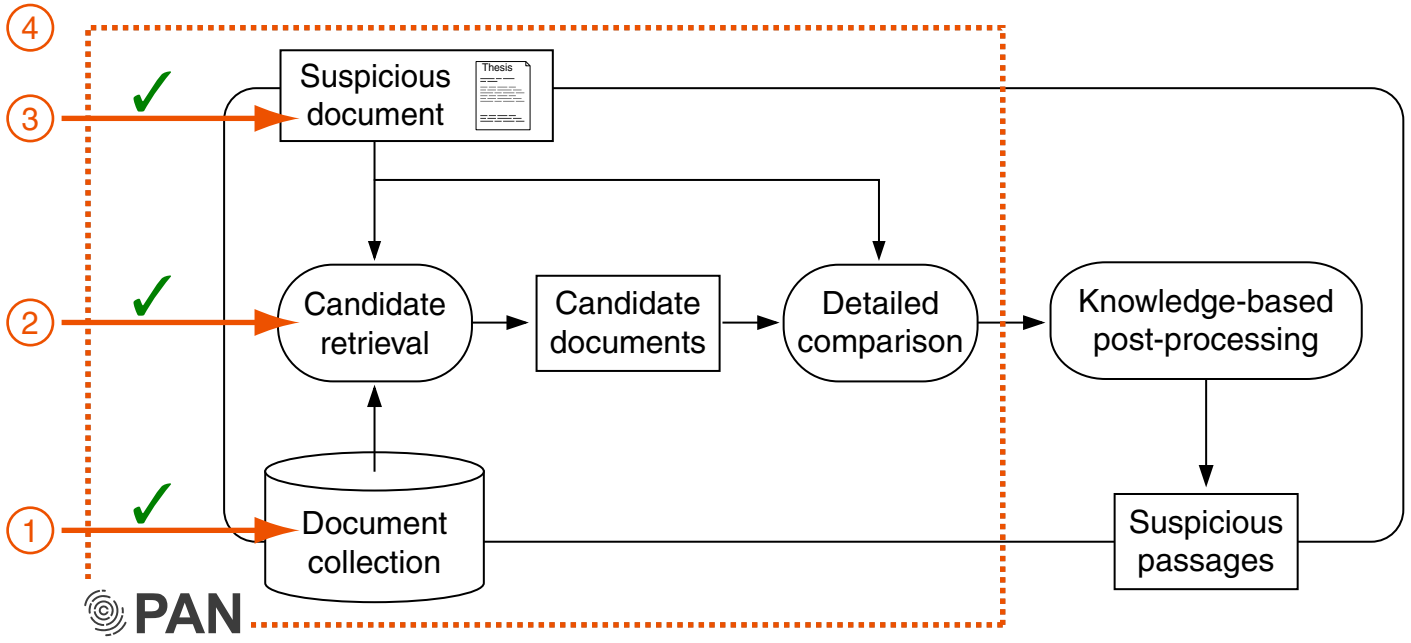
Candidate Retrieval



Considerations:

1. PAN'12 employed the English part of the ClueWeb09 corpus (used in TREC 2009-11 for several tracks) as a static Web snapshot. Size: 500 million web pages, 12.5TB
2. Participants was given efficient corpus access via the API of the ChatNoir search engine. ClueWeb and ChatNoir ensured experiment reproducibility and controllability.
3. The new corpus: manually written digestible texts, topically matching plagiarism cases, Web as source (for document synthesis and plagiarism detection).

Candidate Retrieval

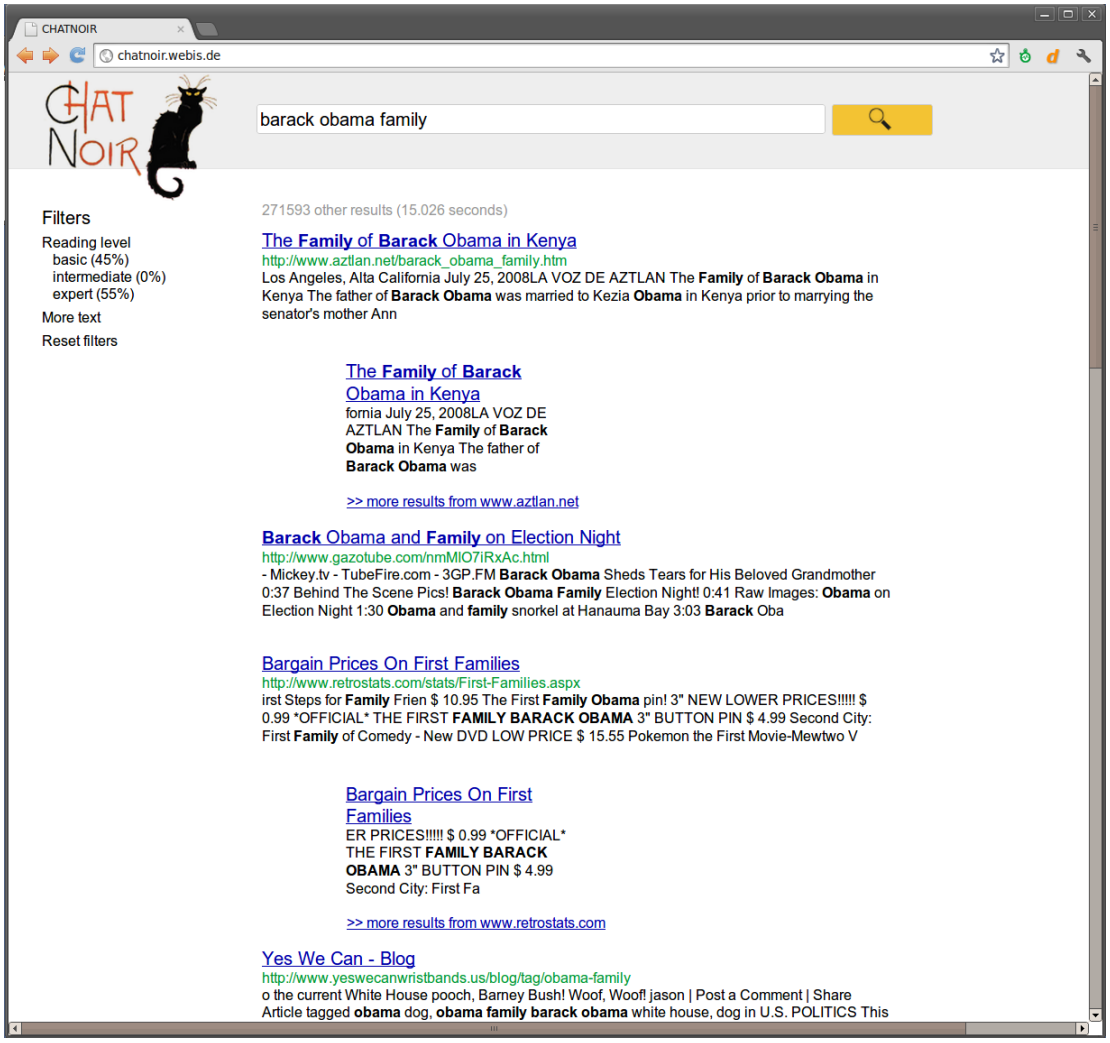


Candidate retrieval task:

- Humans write essays on given topics, plagiarizing from the ClueWeb, using the ChatNoir search engine for research.
- Detectors use ChatNoir to retrieve candidate documents from the ClueWeb.
- Detectors are expected to maximize recall, but use ChatNoir in a cost-effective way.

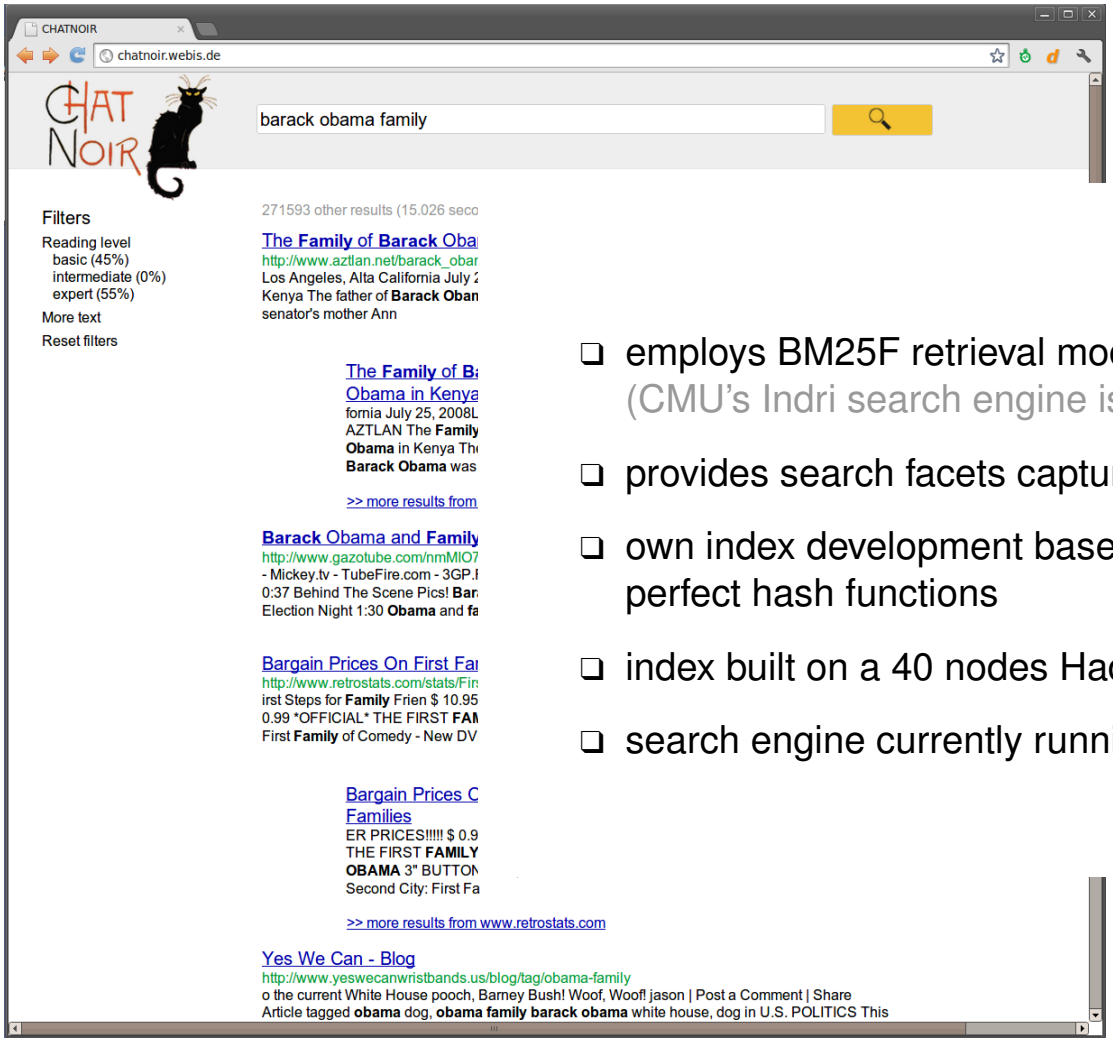
Candidate Retrieval

About ChatNoir [chatnoir.webis.de]



Candidate Retrieval

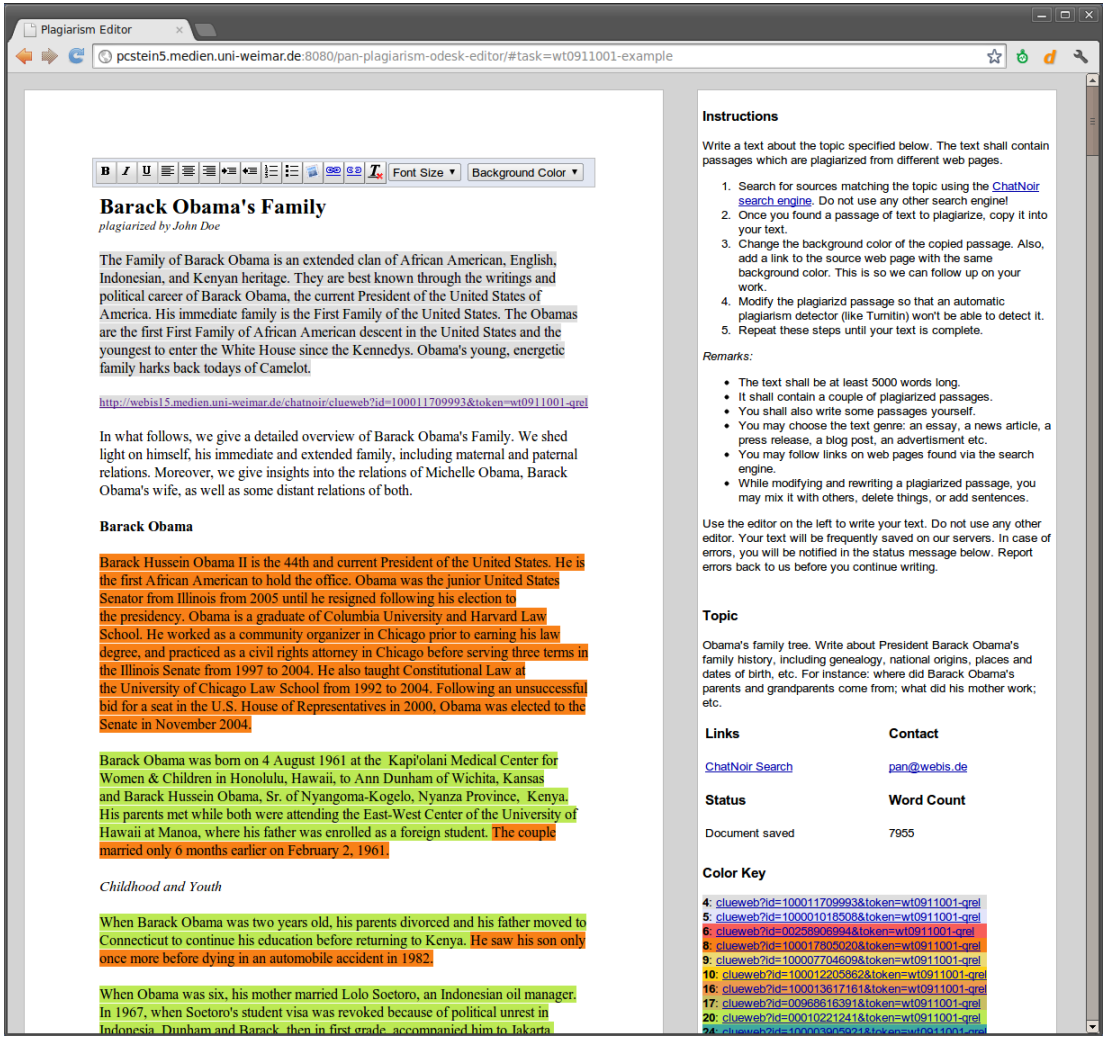
About ChatNoir [chatnoir.webis.de]



- employs BM25F retrieval model
  (CMU's Indri search engine is language-model-based)
- provides search facets capturing readability issues
- own index development based on externalized minimal perfect hash functions
- index built on a 40 nodes Hadoop cluster
- search engine currently running on 11 machines

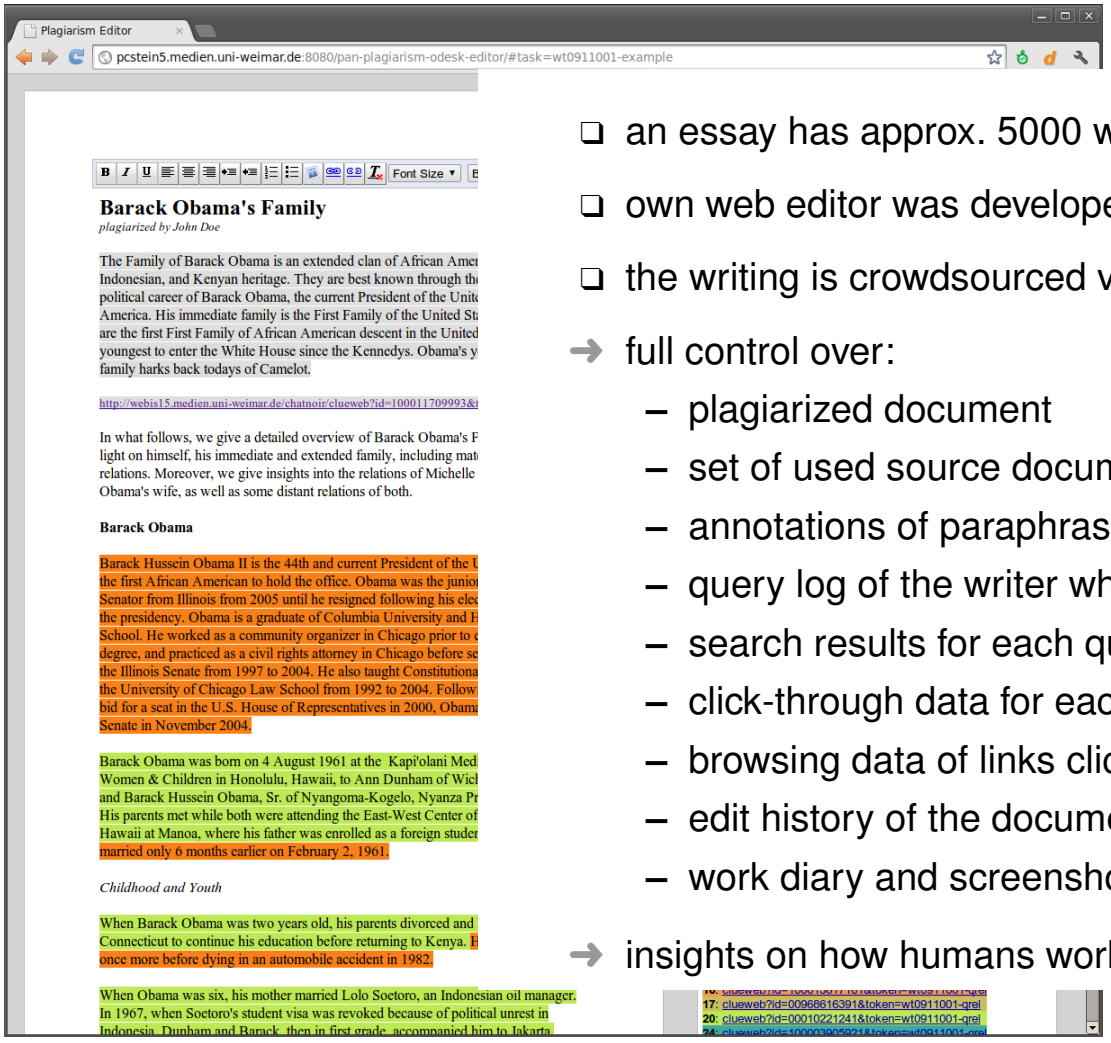
Candidate Retrieval

About Corpus Construction



Candidate Retrieval

About Corpus Construction



- ❑ an essay has approx. 5000 words which means 8-10 pages
- ❑ own web editor was developed for essay writing
- ❑ the writing is crowdsourced via oDesk

➜ full control over:

- plagiarized document
- set of used source documents
- annotations of paraphrased passages
- query log of the writer while researching the topic
- search results for each query
- click-through data for each query
- browsing data of links clicked within ClueWeb
- edit history of the document covering all keystrokes
- work diary and screenshots as recorded by oDesk

➜ insights on how humans work when reusing text

Candidate Retrieval

Survey of Approaches

An analysis of the participants' notebooks reveals a candidate retrieval process:

1. Chunking
   Given a suspicious document, it is divided into (possibly overlapping) passages of text. Each chunk of text is then processed individually.
2. Keyphrase Extraction
   Given a chunk (or the entire suspicious document), keyphrases are extracted from it in order to formulate queries with them.
3. Query Formulation
   Given sets of keywords extracted from chunks, queries are formulated which are tailored to the API of the search engine used.
4. Search Control
   Given a set of queries, the search controller schedules their submission to the search engine and directs the download of search results.
5. Download Filtering
   Given a set of downloaded documents, all documents are removed that are not worthwhile for detailed comparison to the suspicious document.

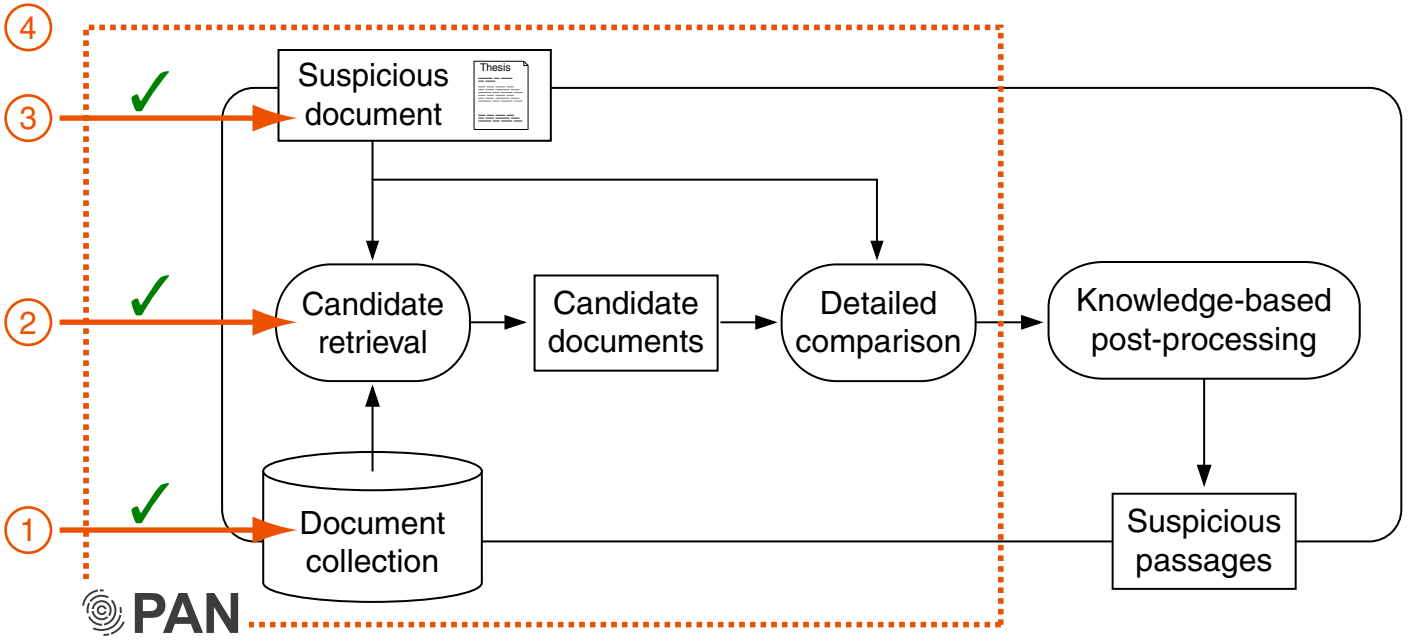
Candidate Retrieval

Evaluation Results

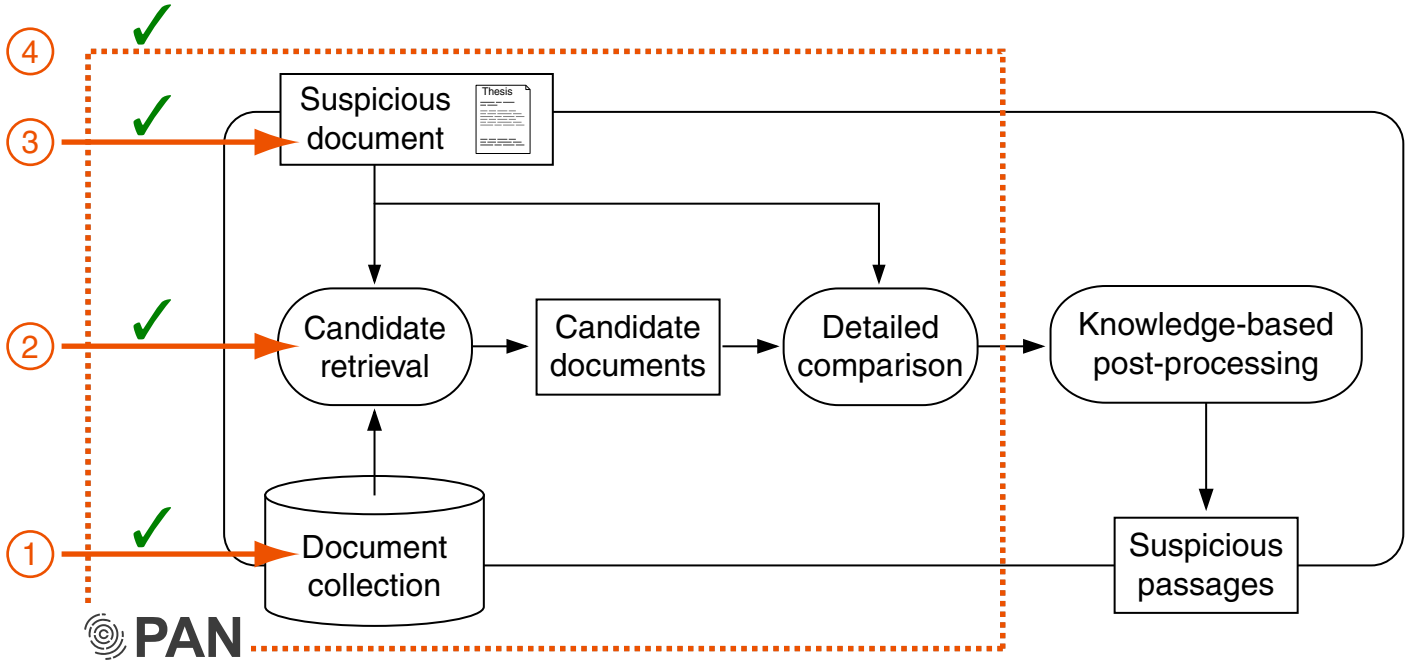
Team	Total Workload		Time to 1st Detection		Reported Sources		Downloaded Sources	
	Queries	Dwnlds	Queries	Dwnlds	Precision	Recall	Precision	Recall
Gillam	63	527	5	26	0.63	0.25	0.01	**0.56**
Jayapal	67	174	9	14	**0.66**	**0.28**	0.07	0.43
Kong	551	327	81	28	0.57	0.24	0.02	0.37
Palkovskii	63	1027	27	319	0.44	0.12	0.00	0.21
Suchomel	**13**	**95**	**6**	2	0.52	0.21	**0.08**	0.35

- Suchomel et al. implement the best tradeoff between cost and quality.
- Jayapal implements the best approach in terms of precision and recall.

Detailed Comparison



Detailed Comparison



Detailed comparison task:

- Detectors are presented with a suspicious and a candidate document, and are asked to extract the plagiarized passages.
- Developers submit their detection softwares instead of detection results.
- This allows for re-evaluating detectors, as well as to measure runtime and to use private corpora.

Detailed Comparison

Software Submissions and Runtime Analysis

- Eleven participants, about the average number from last years.

➜ Software submissions do not distract people from participating.

Team	Submission Size [MB]	Operating System	Programming Language	Average Runtime [sec/comparison]
Rodríguez Torrejón	1.80	Linux	sh, C/C++	0.19
Sánchez-Vega	0.04	Linux	C++	2.48
Oberreuter	0.19	Linux	Java	2.58
Palkovskii	68.20	Windows	C#	4.51
Grozea	1.90	Linux	Perl, Octave	4.82
Suchomel	0.02	Linux	Perl	5.36
Kong	2.60	Linux	Java	5.91
Jayapal	37.20	Linux	Java	8.43
Gillam	0.48	Linux	Python 2.7	9.40
Küppers	42.90	Linux	Java	27.64
Ghosh	554.50	Linux	sh, Java	–

➜ Congratulations to Rodríguez Torrejón et al. for submitting the most efficient detailed comparison program.

Detailed Comparison

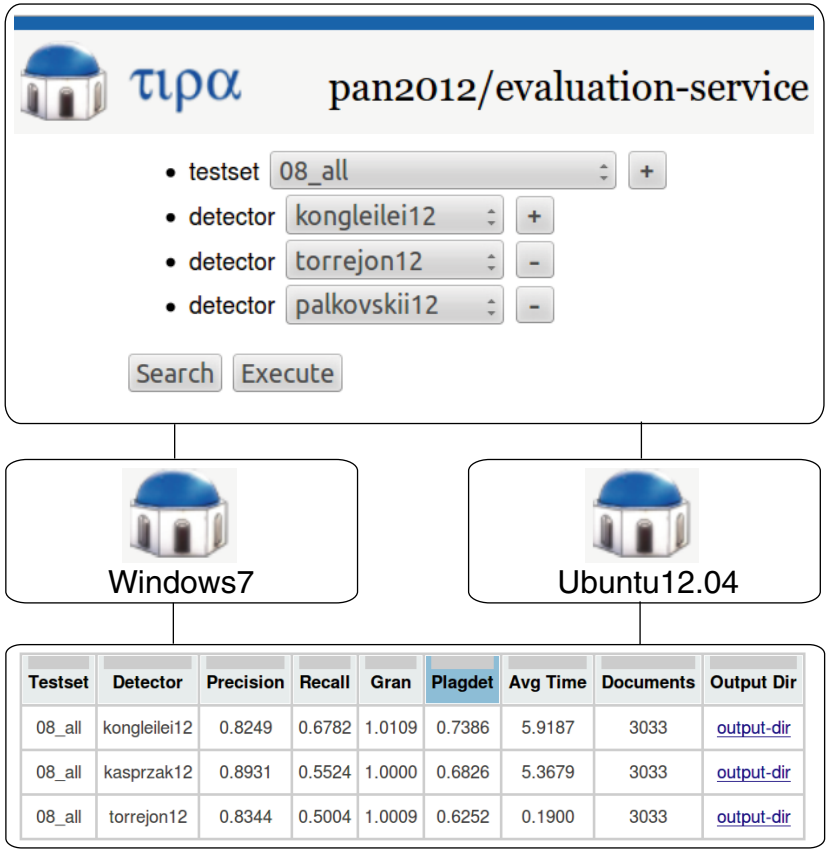
Survey of Approaches

An analysis of the participants' notebooks reveals a detailed comparison process:

1. Seeding
   Given a suspicious document and a source document, matches (also called „seeds") between the two documents are identified using some seed heuristic. Seed heuristics either identify exact matches or *create* matches by changing the underlying texts in a domain-specific or linguistically motivated way.
2. Match Merging
   Given seed matches identified between a suspicious document and a source document, they are merged into aligned text passages of maximal length between the two documents which are then reported as plagiarism detections.
3. Passage Filtering
   Given a set of aligned passages, a passage filter removes all aligned passages that do not meet certain criteria.

Detailed Comparison

TIRA evaluation platform



Testset	Detector	Precision	Recall	Gran	Plagdet	Avg Time	Documents	Output Dir
08_all	kongleilei12	0.8249	0.6782	1.0109	0.7386	5.9187	3033	output-dir
08_all	kasprzak12	0.8931	0.5524	1.0000	0.6826	5.3679	3033	output-dir
08_all	torrejon12	0.8344	0.5004	1.0009	0.6252	0.1900	3033	output-dir

[tira@localhost] [tira@buw]

- TIRA takes locally executable programs and turns them into web services.
- TIRA assumes responsibility for storing and indexing of execution results.
- For the PAN evaluation, TIRA servers are provided for two operating systems, Windows and Ubuntu.
- Participants submit their plagiarism detection software for deployment on the appropriate TIRA server.
- A third TIRA server controls the overall evaluation of all deployed submissions on the private test set and provides the overall results.

Detailed Comparison

Evaluation Corpus Construction

- Like in last years based on books from Project Gutenberg.
- Divided into seven sub-corpora:

Evaluation Corpus Statistics

Sub-Corpus	Number of Cases	Avg. Cosine Similarity
Real Cases	33	0.161
Simulated	500	0.364
Translation ($\{$de, es$\} \rightarrow$ en)	500	0.018
Artificial (High)	500	0.392
Artificial (Low)	500	0.455
No Obfuscation	500	0.560
No Plagiarism	500	0.431
Overall	3033	0.369

- Similarity of document pairs was taken into account this year.
- Real Cases were taken from the Web. Cross-Language cases were constructed using the multi-lingual Europarl corpus.

Detailed Comparison

Evaluation Results: Overall Performance

Rank / Team	PlagDet	Precision	Recall	Granularity
1 Kong	**0.738**	0.824	**0.678**	1.01
2 Suchomel	0.682	0.893	0.552	**1.00**
3 Grozea	0.678	0.774	0.635	1.03
4 Oberreuter	0.673	0.867	0.555	**1.00**
5 Rodríguez Torrejón	0.625	0.834	0.500	**1.00**
6 Palkovskii	0.538	0.574	0.523	1.02
7 Küppers	0.349	0.776	0.282	1.26
8 Sánchez-Vega	0.309	0.537	0.349	1.57
9 Gillam	0.308	**0.898**	0.190	1.02
10 Jayapal	0.045	0.622	0.075	6.93

➜ Congratulations to Kong et al. for submitting the most effective detailed comparison program.

Summary and Outlook

PAN 2012:

- Task-wise evaluation of plagiarism detectors.
- Candidate document retrieval at Web scale using ChatNoir.
- Software submissions for sustainable / repeatable evaluation using TIRA.
- More realistic plagiarism corpus.
- New performance measures in addition to the traditional ones.

➜ A lot of fun!
Thanks to everyone who volunteered to test our new setup!

PAN 2013 and beyond:

- Improvement and consolidation of the new tools.
- Use of the plagiarism corpus for detailed comparison as well.
- Community process to collect more plagiarism (real and manual).

➜ Fully automatic plagiarism detection evaluations.