Detecting Automatically Generated Sentences with Grammatical Structure Similarity

Tien M Nguyen
Cyril Labbé

University Grenoble Alpes

minh-tien.nguyen@univ-grenoble-alpes.fr
cyril.labbe@univ-grenoble-alpes.fr

April 8, 2017

Introduction

- Automatic scientific paper generators exist and they have been used to manipulate index score
  [Labbe, 2010, Beel and Gipp, 2010, López-Cózar et al., 2013] or publication bibliography[Noorden, 2014].
- Detection methods are needed for both publisher and online archive.

Introduction

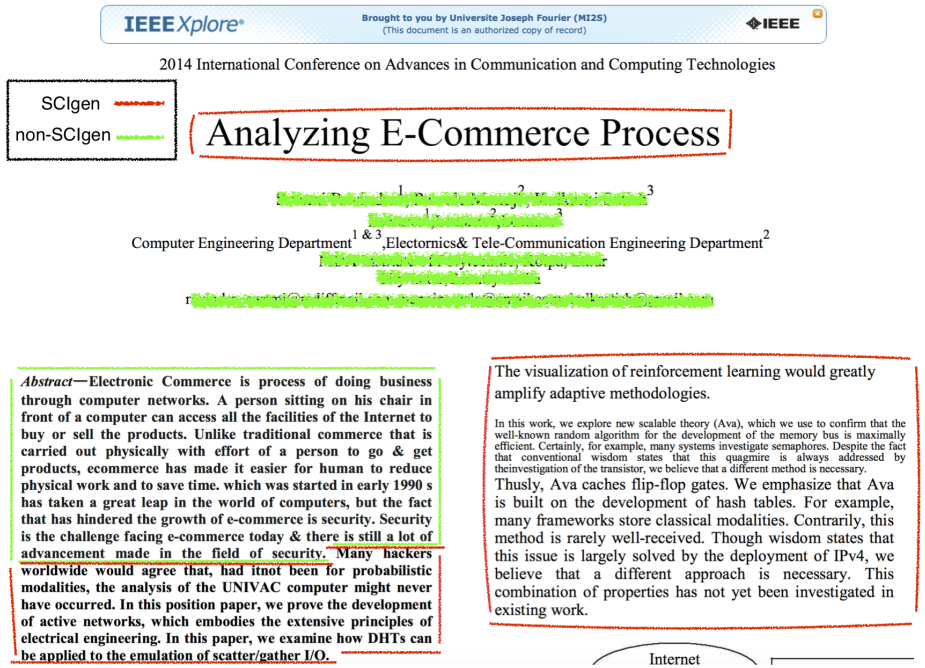
- Automatic scientific paper generators exist and they have been used to manipulate index score
  [Labbe, 2010, Beel and Gipp, 2010, López-Cózar et al., 2013] or publication bibliography[Noorden, 2014].
- Detection methods are needed for both publisher and online archive.

- Checking the reference section [Xiong and Huang, 2009] or ad-hoc similarity measure with custom weights
  [Lavoie and Krishnamoorthy, 2010].
- [Williams and Giles, 2015] proposed a pseudo-relevance feedback method.
- [Amancio, 2015] made use of complex networks to obtained a Scigen discrimination rate of at least 89%.
- [Nguyen and Labbé, 2016] with textual distance achieved a very good result.

Introduction

Problem

- Current methods are all at the document level. Unable to deal with short segments of automatically generated text.



- Current generators use Probabilistic context free grammar (PCFG) to generate phrases, sentences, structure, etc.

Outline

Outline

Probabilistic Context Free Grammar (FCFG)

- Know generators make use of PCFG including: SCIgen, SCIgen-Physic(Physgen), Mathgen, Automatic SBIR Proposal Generator(Propgen).
- PCFG use sets of rules for generation.

Example

parts of a rule for a noun phrase NP generation:
NP → a novel SCI_SYSTEM for the SCI_ACT.
SCI_ACT → SCI_ACT_A SCI_THING| SCI_ACT_A SCI_THING that SCI_EFFECT
SCI_ACT_A → understanding of| SCI_ADJ unification of SCI_THING and| SCI_VERBION of
SCI_SYSTEM → algorithm| system| framework| heuristic| application| SCI_APPROACH
SCI_THING → IPv4| IPv6 | telephony| multi-processors| compilers | semaphores| RPCs| etc...

SCI_VERBION → exploration| development| refinement| investigation| analysis | etc...

Using this rule, these phrases can be generated:
- a novel heuristic for the understanding of randomized algorithms
- a novel system for the typical unification of massive multiplayer online role-playing games and symmetric encryption
- a novel framework for the development of scatter/gather I/O
- a novel system for the analysis of gigabit switches

Outline

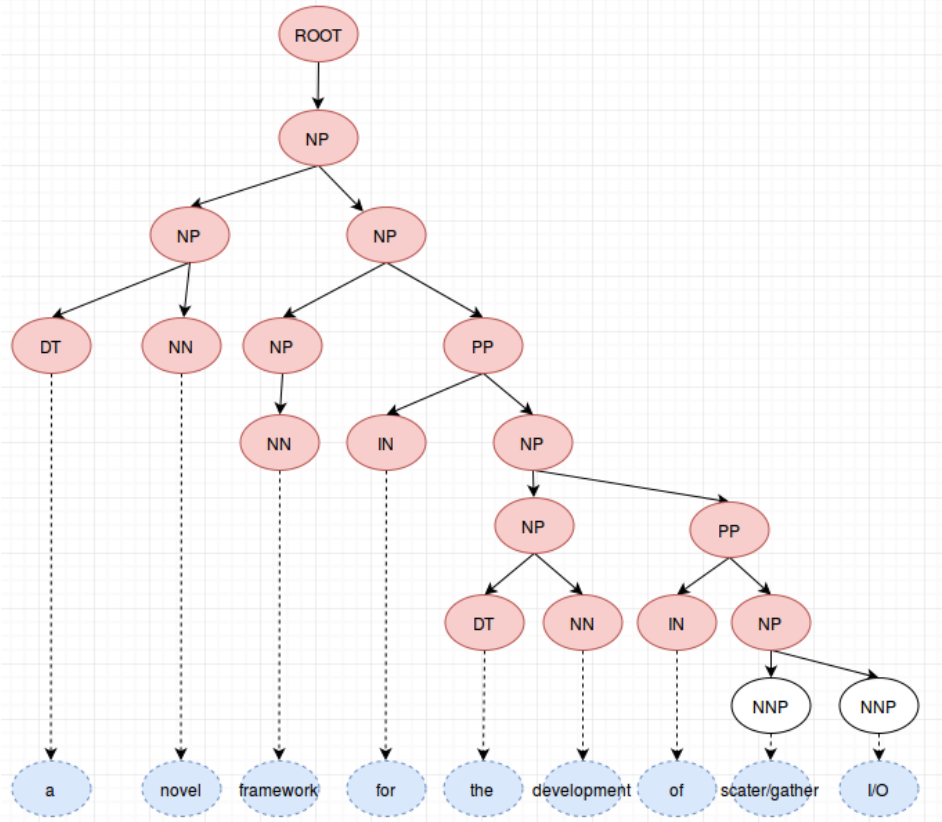
Using Parse Structure Tree On Sentence Similarity

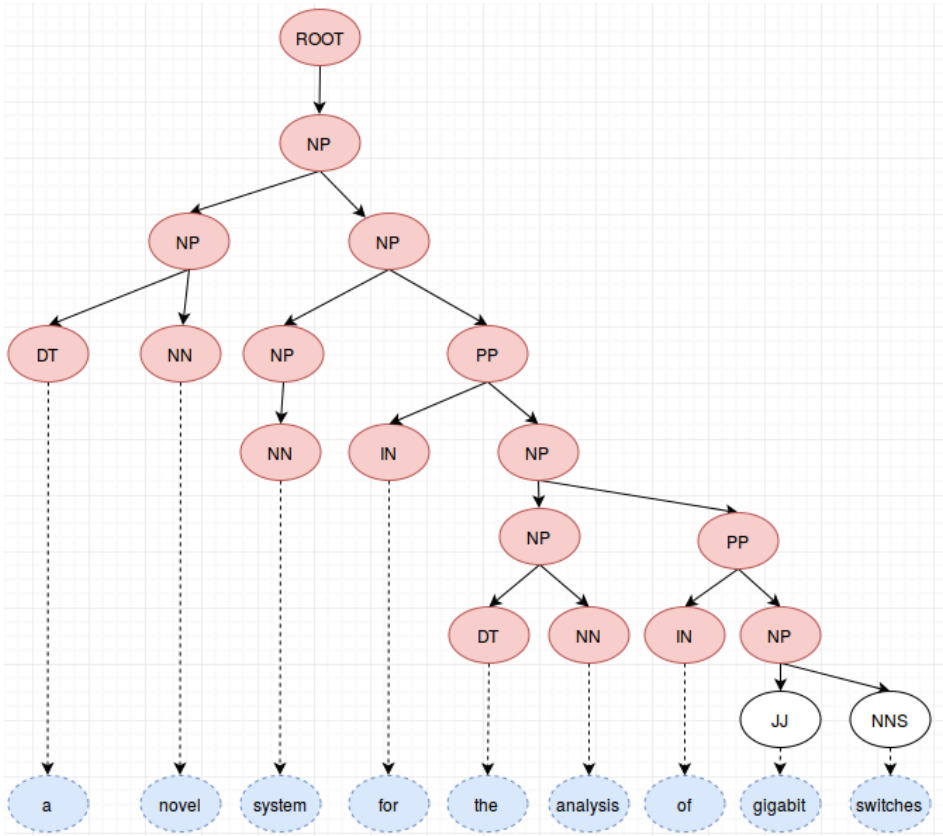
Parse structure tree or parse tree is a tree that represents the syntactic structure of a sentence or a phrase.

Using Parse Structure Tree On Sentence Similarity

Parse structure tree or parse tree is a tree that represents the syntactic structure of a sentence or a phrase.

Parse trees from previous example.



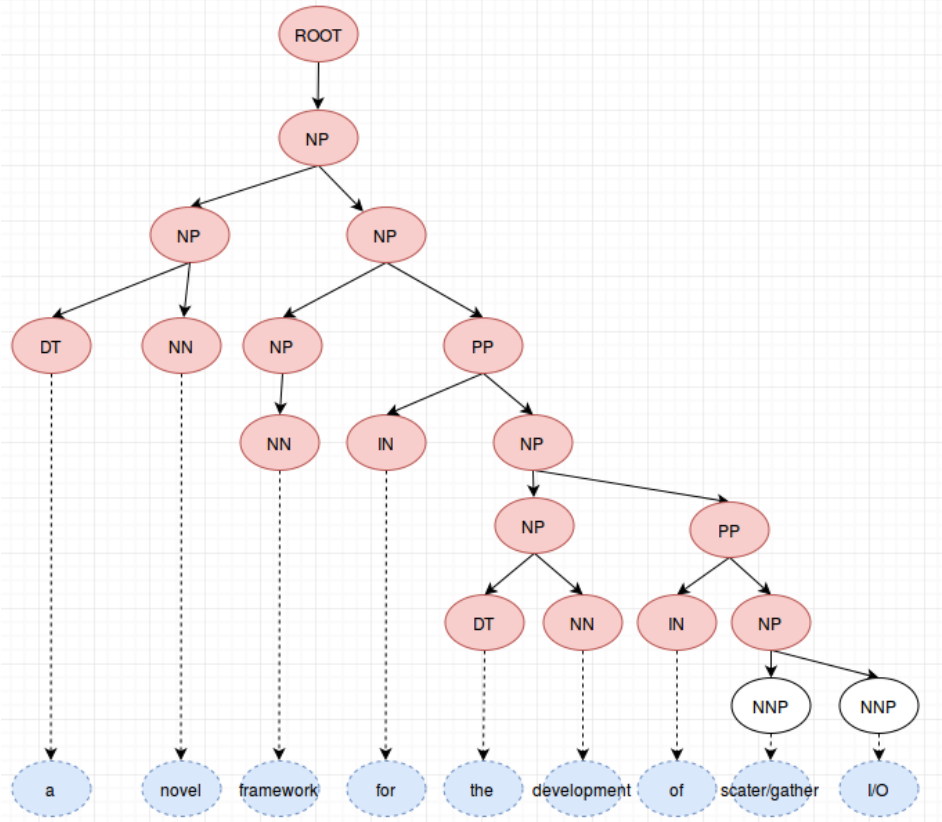


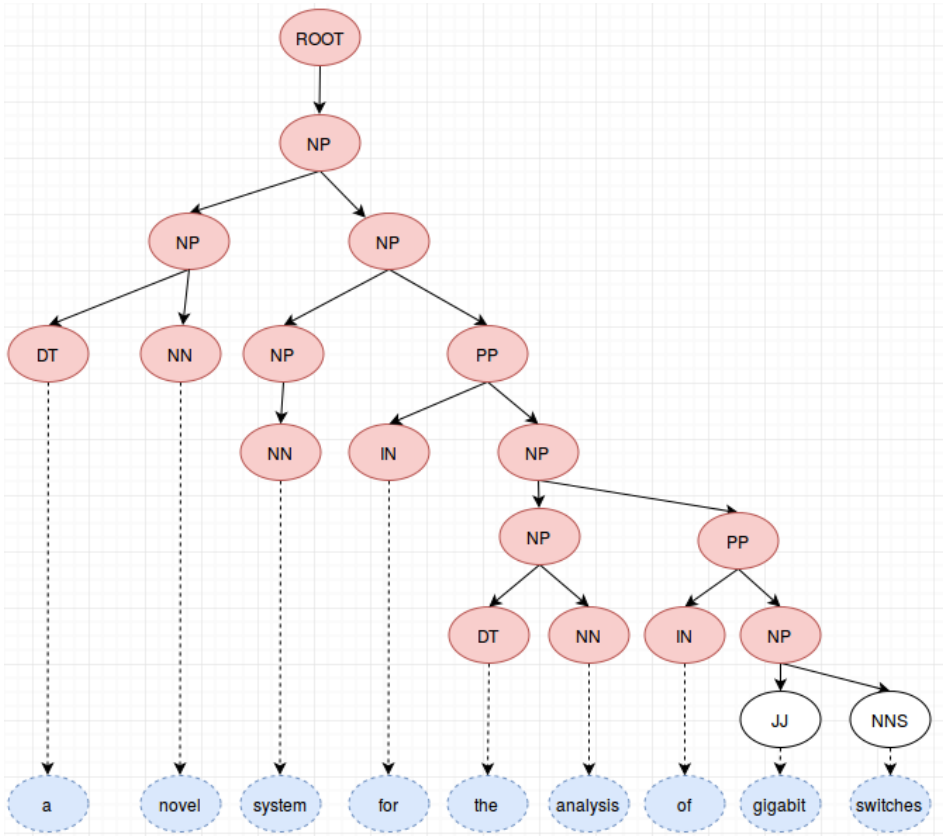
.

Using Parse Structure Tree On Sentence Similarity

Parse structure tree or parse tree is a tree that represents the syntactic structure of a sentence or a phrase.

Parse trees from previous example.





.

Similar sentences have similar parse tree structure.

Using Parse Tree On Sentence Similarity

- [Culotta and Sorensen, 2004] proposed to estimate sentences similarity using augmented parse trees.
- DLSITE-2 system [Wang and Neumann, 2007] was used to determine the textual entailment using syntactic parse tree.
- Parse tree with similar words were used by [Durán et al., 2014] to determine the similarity between sentences.
- [Zubarev and Sochenkov, 2014] used their sentence similarity measurement in Exactus [Sochenkov et al., 2016] to detect plagiarism.

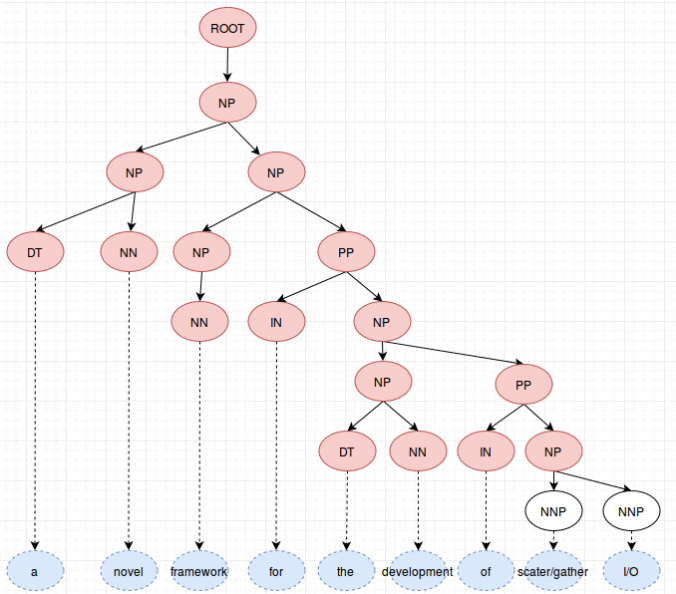
Using Parse Tree On Sentence Similarity

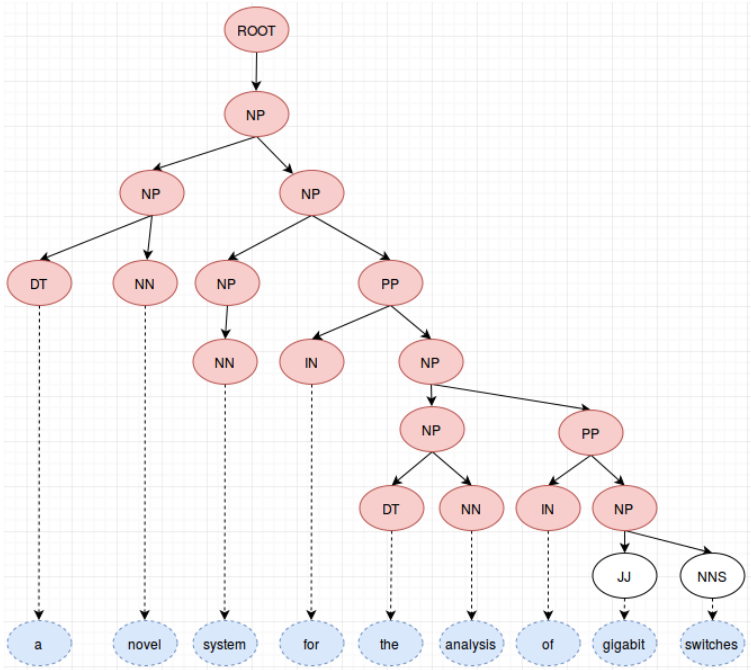
- [Culotta and Sorensen, 2004] proposed to estimate sentences similarity using augmented parse trees.
- DLSITE-2 system [Wang and Neumann, 2007] was used to determine the textual entailment using syntactic parse tree.
- Parse tree with similar words were used by [Durán et al., 2014] to determine the similarity between sentences.
- [Zubarev and Sochenkov, 2014] used their sentence similarity measurement in Exactus [Sochenkov et al., 2016] to detect plagiarism.
- However they are either too expensive or the similarity measure is too focus on pairs of common words so might not fit our need.
- We will present the GSS system based on parse tree.

Outline

Pre-treatment

- Texts are normalize and split to sentences. A parse tree for each sentence is created using Stanford Parser [Marneffe and Manning, 2008].
- Lemmas are removed from the tree, only the structure node are kept.





Grammatical Structure Similarity

Definition

Grammatical Structure Similarity (GSS):
Let N_A be the number of node in the parse tree T_A of sentence A, N_B be the number of node in the parse tree T_B of sentence B, and N_{AB} be the number of node in the biggest common subtree of T_A and T_B. Then the Grammatical Structure Similarity between A and B is defined as:

$$GSS_{(A,B)} = \frac{2*N_{AB}}{N_A+N_B}$$

GSS between previous example $GSS_{(F_{1A}/F_{1B})} = \frac{2*17}{19+19} = 0.89$

Grammatical Structure Similarity

Definition

Grammatical Structure Similarity (GSS):
Let N_A be the number of node in the parse tree T_A of sentence A, N_B be the number of node in the parse tree T_B of sentence B, and N_{AB} be the number of node in the biggest common subtree of T_A and T_B. Then the Grammatical Structure Similarity between A and B is defined as:

$$GSS_{(A,B)} = \frac{2*N_{AB}}{N_A+N_B}$$

GSS between previous example $GSS_{(F_{1A}/F_{1B})} = \frac{2*17}{19+19} = 0.89$

Definition

Maximum Grammatical Structure Similarity (MGSS): For a sentence A in the test corpus (C_T), The MGSS between A and the PCFG corpus (C_{PCFG}) is:

$$MGSS_{(A,C_{PCFG})} = Max_{(B \in C_{PCFG})}(GSS_{A,B})$$

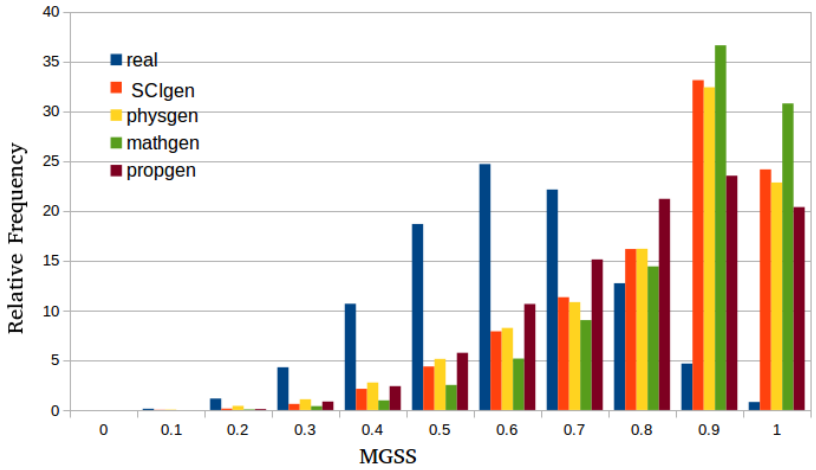
GSS and PCFG Corpus Effectiveness

- Test corpus: 100 texts from each generator plus 100 genuine written texts.
- PCFG copora

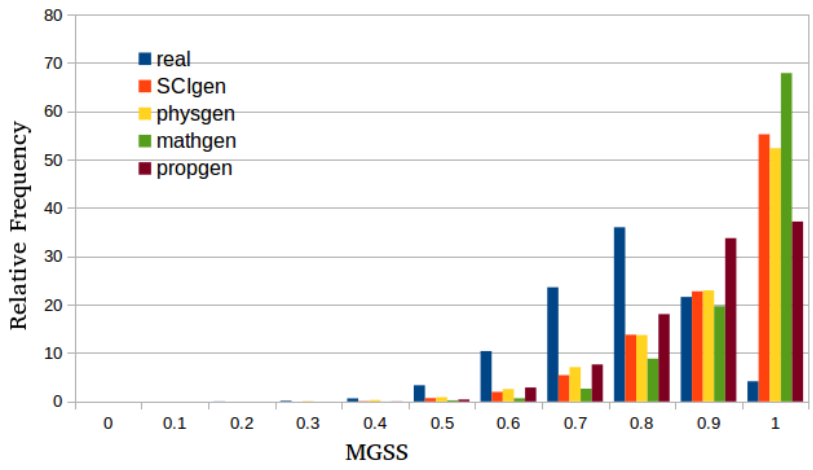
copus size	nb of sentence	nb of distinct sentence	nb of distinct parse tree
80	12.1k	9.2k	8k
160	25.5k	18.2k	14.8k
320	45.5k	33.2k	26.9k

- Even though the number of distinct sentence as well as parse tree seem quite numerous, we believe that most of them should also be somewhat similar to each other.
- The size of the PCFG corpus affect processing time.

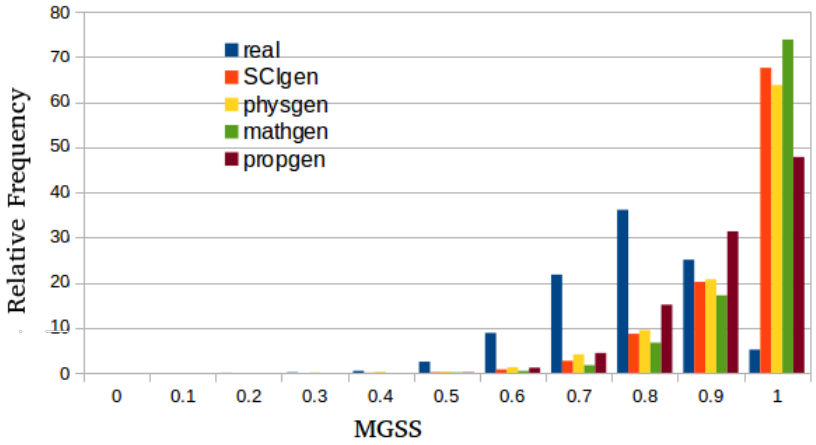
GSS and PCFG Corpus Effectiveness



PCFG corpus with 80 documents

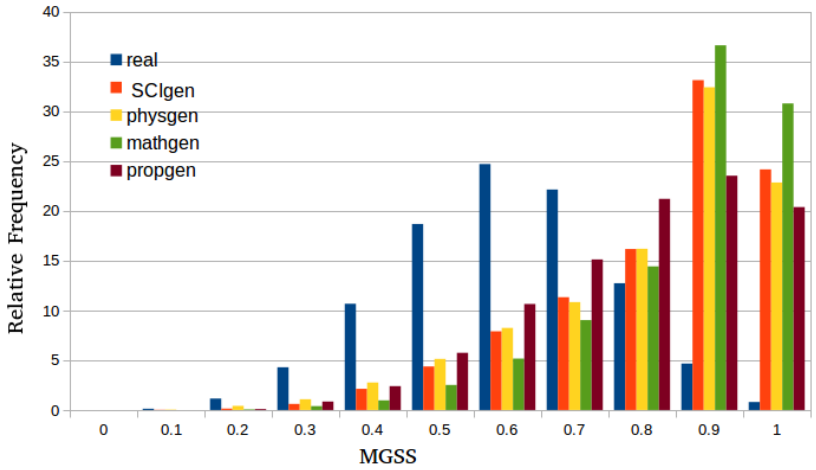


PCFG corpus with 160 documents

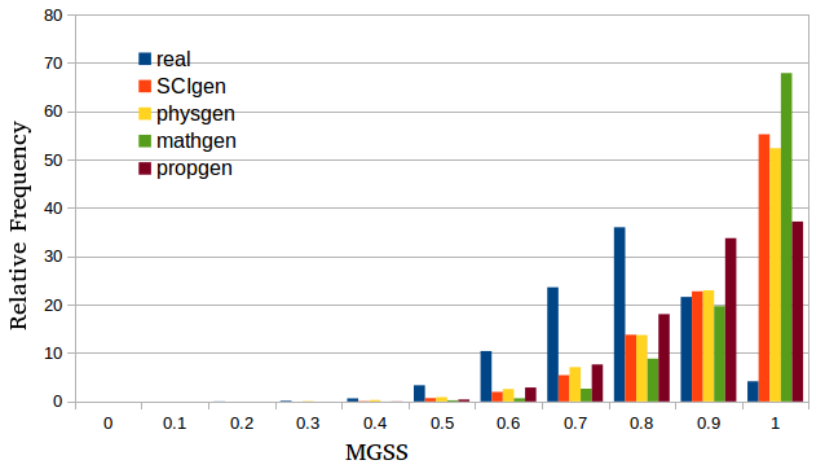


PCFG corpus with 320 documents

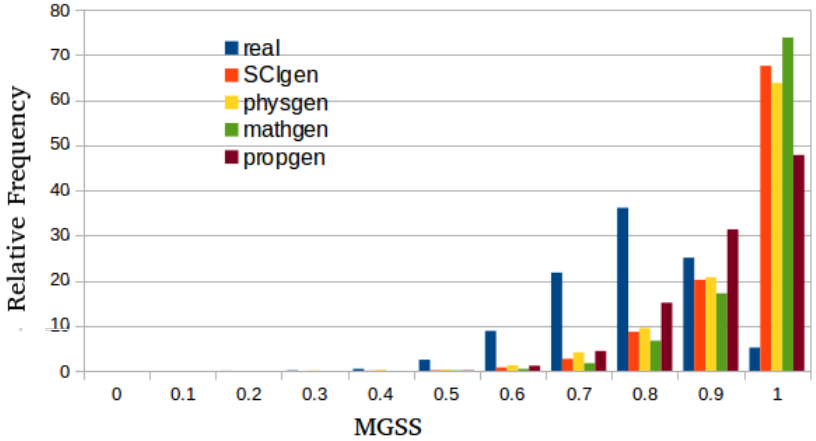
GSS and PCFG Corpus Effectiveness



PCFG corpus with 80 documents



PCFG corpus with 160 documents

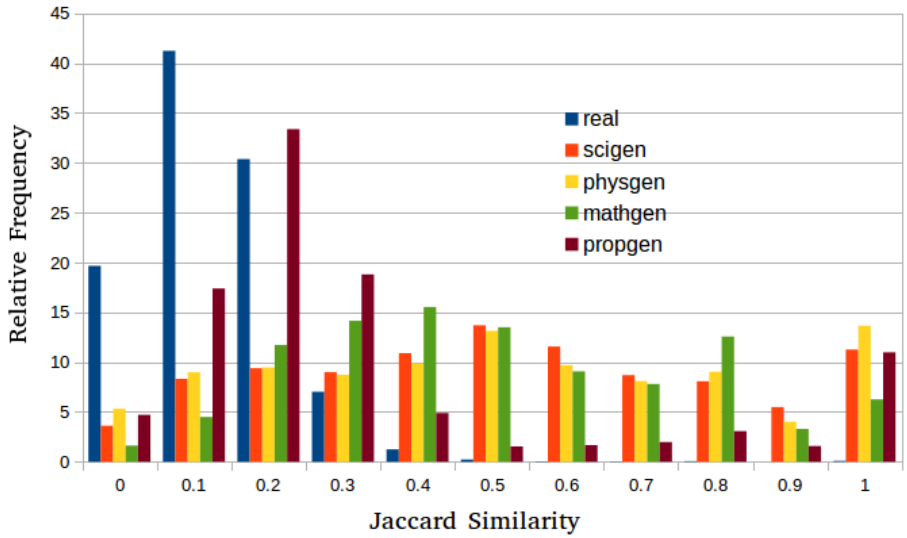


PCFG corpus with 320 documents

PCFG corpus with 160 generated texts seems to be the most balance.

A Filter With Jaccard similarity

- The cost for parsing is quite expensive.
- A filter to reduce the number of sentence that need to be parsed.
- The Figure shows the Jaccard similarity (the number of common word over the total number of distinct word in two sentences) over the test and FCPG sample corpora.



- If the threshold is set at 0.3, only 10-20% of genuine written sentences need to be parsed compare to 80% of generated sentences.

Some typical mistakes

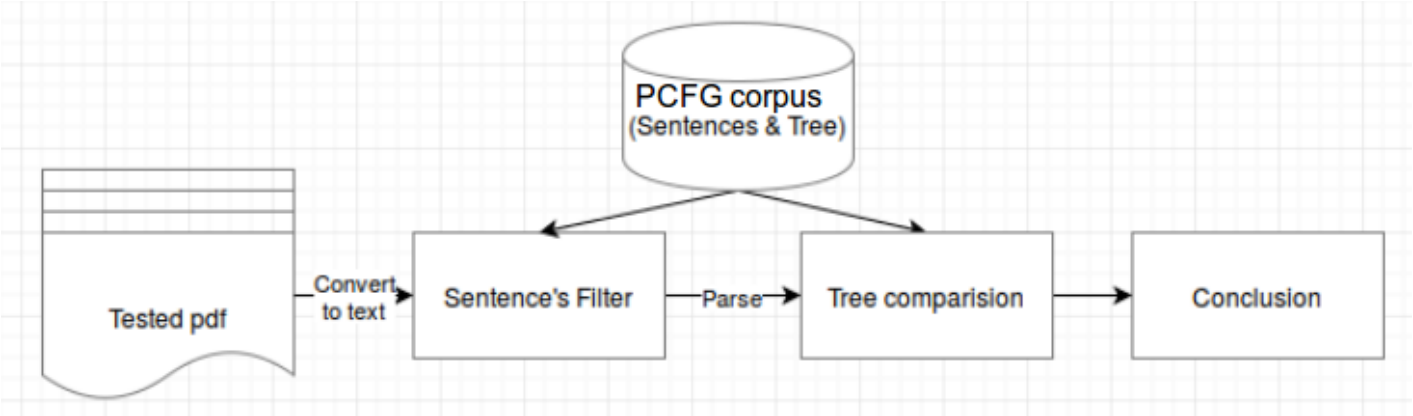
- However some common sentences were used in automatically generators.

Genuine written sentence	Sentence in sample corpus	Jaccard similarty	GSS
our main contributions are as follows	our main contributions are as follows	1	1
it is easy to see that	it is easy to see that	1	1
the states of this network are	the contributions of this work are as follows	0.4	0.89
the rest of the paper is organized as follows	the rest of this paper is organized as follows	0.88	1
the remainder of the paper is organized as follows	the rest of this paper is organized as follows	0.8	1
the proof of the claim can be found in appendix	useful survey of the subject can be found in	0.58	0.9
the interpretation of the walk is as follows	the rest of the paper proceeds as follows	0.45	1

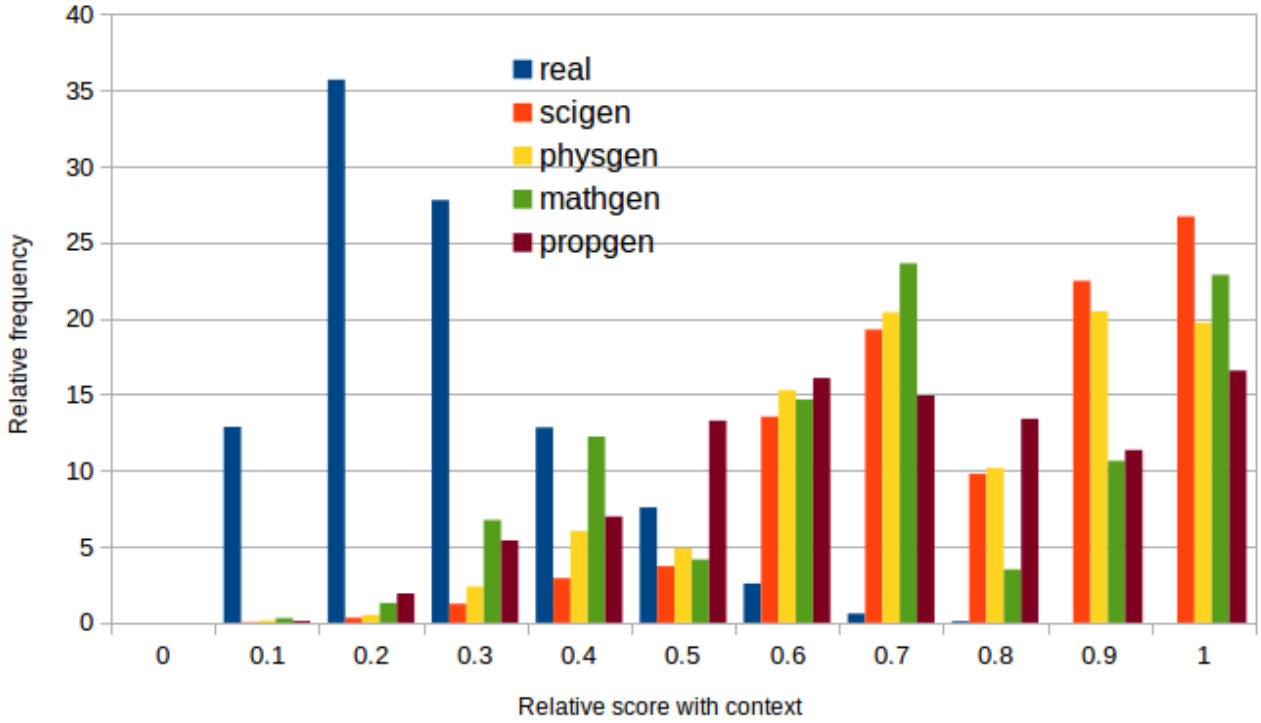
Outline

GSS system with Jaccard Filter and Context

- Our aim is to detect a small portion of automatically generated text.
- We consider each sentence along with its context which includes the **direct previous** and **next sentence** to balance out special cases.
- Jaccard similarity filter at 0.3 to reduce computational need.
- PCFG sample corpus of 160 generated papers.



Test Results



- This graph only shows the result of GSS system with sentences that passed the filter.
- 96.7% genuine written sentence with less than 0.5 GSS.
- From 70% to more than 90% of automatically generated sentence with higher than 0.5 GSS.

Outline

Comparison With Other Methods

- Comparative results against different machine learning techniques.
- Using PCFG sample corpora as shown earlier and counterpart real corporal of the same size
- Sentences are label as "real" or "generated" with remove spare terms at 0.98
- Possibility of detecting modified version of a known generator (namely possibility of detecting Physgen papers using only Scigen as sample.)
- Thresholds for both GSS methods were set at 0.5.

Comparison With Other Methods

- False positive rate: the probability a genuine written sentence is marked as generated.
- False negative rate: the probability a generated sentence is marked as genuine.
- Smaller number means less mistakes.

	False positive rate				False negative rate			
corpus size / algorithm	80	160	320	40 Scigen	80	160	320	40 Scigen
Glmnet	0.03	0.02	0.04	0.03	0.80	0.87	0.80	0.63
Maxentropy	0.19	**0.13**	0.20	0.14	0.55	**0.62**	0.52	0.45
SLDA	0.01	0.01	0.05	0.04	0.95	0.97	0.78	0.63
Boosting	0.58	**0.5**	0.62	0.14	0.23	**0.11**	0.21	0.49
Bagging	0.10	**0.05**	0.1	0.06	0.37	**0.5**	0.35	0.42
Random Forest	0.07	**0.05**	0.07	0.05	0.49	**0.58**	0.43	0.44
Tree	0.02	0.02	0.02	0.03	0.87	0.88	0.87	0.65
GSS only	0.85	0.95	0.97	0.73	0.07	0.007	0.002	0.015
GSS system	0.008	**0.008**	0.01	0.002	0.25	**0.19**	0.17	0.21

Comparison With Other Methods

- False positive rate: the probability a genuine written sentence is marked as generated.
- False negative rate: the probability a generated sentence is marked as genuine.
- Smaller number means less mistakes.

	False positive rate				False negative rate			
corpus size / algorithm	80	160	320	40 Scigen	80	160	320	40 Scigen
Glmnet	0.03	0.02	0.04	0.03	0.80	0.87	0.80	0.63
Maxentropy	0.19	**0.13**	0.20	0.14	0.55	**0.62**	0.52	0.45
SLDA	0.01	0.01	0.05	0.04	0.95	0.97	0.78	0.63
Boosting	0.58	**0.5**	0.62	0.14	0.23	**0.11**	0.21	0.49
Bagging	0.10	**0.05**	0.1	0.06	0.37	**0.5**	0.35	0.42
Random Forest	0.07	**0.05**	0.07	0.05	0.49	**0.58**	0.43	0.44
Tree	0.02	0.02	0.02	0.03	0.87	0.88	0.87	0.65
GSS only	0.85	0.95	0.97	0.73	0.07	0.007	0.002	0.015
GSS system	0.008	**0.008**	0.01	0.002	0.25	**0.19**	0.17	0.21

- Possibility of detecting Physgen papers using only SCIgen example.

Outline

Conclusion

- Current methods are all at the document level so we propose a method to detect automatically generated sentences based on parse tree.
- Our method achieved true detection rate higher than 80% with less than 1% false detection rate.
- Our method was tested against common machine learning method and proven to give the best result
- Our method also proven the possibility of detecting a modified version of known generators.
- However our method would be ineffective against different generation methods or unknown generators.

References I

Amancio, D. R. (2015).
Comparing the topological properties of real and artificially generated scientific manuscripts.
Scientometrics, 105(3):1763–1779.

Beel, J. and Gipp, B. (2010).
Academic search engine spam and google scholars resilience against it.
Journal of Electronic Publishing.

Culotta, A. and Sorensen, J. (2004).
Dependency tree kernels for relation extraction.
In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

References II

Durán, K., Rodríguez, J., and Bravo, M. (2014).
Similarity of sentences through comparison of syntactic trees with pairs of similar words.
In *Electrical Engineering, Computing Science and Automatic Control (CCE), 2014 11th International Conference on*, pages 1–6.

Labbe, C. (2010).
Ike Antkare one of the great stars in the scientific firmament.
ISSI Newsletter, 6(2):48–52.

Lavoie, A. and Krishnamoorthy, M. (2010).
Algorithmic detection of computer generated text.
arXiv preprint arXiv:1008.0706.

References III


López-Cózar, E. D., Robinson-Garcia, N., and Torres-Salinas, D. (2013).
The google scholar experiment: how to index false papers and manipulate bibliometric indicators.
CoRR, abs/1309.2413.


Marneffe, M.-C. D. and Manning, C. D. (2008).
Stanford typed dependencies manual.


Nguyen, M. and Labbé, C. (2016).
Engineering a tool to detect automatically generated papers.
In *Proceedings of the Third Workshop on Bibliometric-enhanced Information Retrieval co-located with the 38th European Conference on Information Retrieval (ECIR 2016), Padova, Italy, March 20, 2016.*, pages 54–62.

References IV


Noorden, R. V. (2014).
Publishers withdraw more than 120 gibberish papers.
Nature News.


Sochenkov, I., Zubarev, D., Tikhomirov, I., Smirnov, I., Shelmanov, A., Suvorov, R., and Osipov, G. (2016).
Exactus like: Plagiarism detection in scientific texts.
In *European Conference on Information Retrieval*, pages 837–840. Springer.


Wang, R. and Neumann, G. (2007).
Recognizing textual entailment using sentence similarity based on dependency tree skeletons.
In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 36–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

References V


Williams, K. and Giles, C. L. (2015).
On the use of similarity search to detect fake scientific papers.
In *Similarity Search and Applications - 8th International Conference, SISAP 2015*, pages 332–338.


Xiong, J. and Huang, T. (2009).
An effective method to identify machine automatically generated paper.
In *Knowledge Engineering and Software Engineering*, pages 101–102.


Zubarev, D. and Sochenkov, I. (2014).
Using sentence similarity measure for plagiarism source retrieval.
In *CLEF (Working Notes)*, pages 1027–1034.