Enhancing Cohesion and Coherence of Fake Text to Improve Believability for Deceiving Cyber Attackers

Prakruthi Karuna, Hemant Purohit, Özlem Uzuner, Sushil Jajodia, Rajesh Ganesan
Center for Secure Information Systems
George Mason University
{pkaruna, hpurohit, ouzuner, jajodia, rganesan}@gmu.edu

Abstract

Ever increasing ransomware attacks and thefts of intellectual property demand cybersecurity solutions to protect critical documents. One emerging solution is to place fake text documents in the repository of critical documents for deceiving and catching cyber attackers. We can generate fake text documents by obscuring the salient information in legit text documents. However, the obscuring process can result in linguistic inconsistencies, such as broken co-references and illogical flow of ideas across the sentences, which can give away the fake document and render it *unbelievable*.

In this paper, we propose a novel method to generate *believable* fake text documents by automatically improving the linguistic consistency of computer-generated fake text. Our method focuses on enhancing syntactic cohesion and semantic coherence across discourse segments. We conduct experiments with human subjects to evaluate the effect of believability improvements in distinguishing legit texts from fake texts. Results show that the probability to distinguish legit texts from believable fake texts is consistently lower than from fake texts that have not been improved in believability. This indicates the effectiveness of our method in generating believable fake text.

1 Introduction

The rise in the number of cyberattacks, such as the WannaCry ransomware attack[1], has put pressure on governments and corporations to protect their intellectual property and critical documents. Traditional cybersecurity solutions such as access-control, firewalls, malware scanners, intrusion detection and prevention technologies are limited in keeping an attacker from stealing information once he penetrates a computer network. Therefore, recent research has focused on content-based cybersecurity solutions for deceiving an attacker (Rowe and Rrushi, 2016; Jajodia et al., 2016; Heckman et al., 2015) who may succeed in gaining access to the network. These solutions generate and deploy documents with fake content (called 'honeyfiles' or 'decoy files') in the data repositories of legit documents for misleading attackers with false information. Fake documents can be either low interaction honeyfiles such as empty documents with similar names as legit documents, or high interaction honeyfiles with believable but non-informative content that can mislead the attackers (Whitham, 2017; Bowen et al., 2009). However, generating fake content that can deceive a human reader and is indistinguishable from legit content is a challenging task. This research investigates a novel linguistics approach to generate high interaction honeyfiles with believable fake text that are capable of eliciting trust.

The state of the art methods for fake text document generation (Rauti and Leppanen, 2017; Whitham, 2017) are broadly categorized based on the nature of content generated as follows: (1) random character generation, (2) generation based on random word and sentence extraction from a given public document corpus, (3) rule-based and preset template-based text generation, (4) generation based on translation from one language to another containing partial content from an existing document, and lastly, (5) generation based on language models built from a collection of similar documents (Whitham, 2017; Voris et al., 2012). However, several of the resulting automatically generated text suffers from lack of believability,

[1] https://www.tripwire.com/state-of-security/security-data-protection/cyber-security/10-significant-ransomware-attacks-2017/

Proceedings of the First International Workshop on Language Cognition and Computational Models, pages 31–40
Santa Fe, New Mexico, United States, August 20, 2018.
https://doi.org/10.18653/v1/P17

i.e. linguistic inconsistencies and disfluencies give it away as fake text. Believability is essential to the success of cyber deception (Voris et al., 2013). Our goal is to automatically generate believable fake documents that can deceive attackers.

The believability of a given fake text for a human reader is difficult to assess (Bowen et al., 2009; McNamara and Kintsch, 1996; Otero and Kintsch, 1992). Believability has two major factors: first, the prior knowledge of a reader (attacker) and second, the characteristics of the text. While prior knowledge can affect the believability of text, such knowledge can vary from attacker to attacker, resulting in different degrees of believability for different attackers. Textual characteristics, on the other hand, can affect believability even for attackers with no prior knowledge. We hypothesize that cohesion and coherence of text are two major factors in this respect.

We define a fake text in this research as a modified version of a legit human-written text created automatically by removing some sentences that contain salient information. We define a believable fake text as the modified version of a fake text with higher cohesion and coherence than the fake text. Prior research provides metrics for measuring cohesion and coherence based on linguistic characteristics of text (McNamara et al., 2014; Lin et al., 2011; Lapata and Barzilay, 2005). Also, the literature on text simplification and summarization provides techniques to improve cohesion and coherence of a given text (Narayan, 2014; Siddharthan et al., 2011; Mani et al., 1999). However, the question of how to effectively manipulate a given text to improve its cohesion and coherence so as to render it believable still requires more investigation.

Our specific research questions are the following: a) how can we adapt existing NLP techniques to automatically modify a given fake text to increase its cohesion and coherence? and b) what is the relation between cohesion, coherence, and believability of a given text for a reader? We study syntactic cohesion at the local sentence level and semantic coherence at the paragraph level. We evaluate our method in two ways. First, we test for a statistically significant increase in the cohesion and coherence of a believable fake text over its corresponding (unbelievable) fake text. Second, we conduct a 'believability test' (Bowen et al., 2009) with human subjects for identifying the legit text from a given pair of legit and believable fake texts. Our results show that the probability to distinguish a legit text against a believable fake text is less than 50%, while that against a (unbelievable) fake text is greater than 50%. These results indicate the effectiveness of our method in generating believable fake texts. Our specific contributions are the following:

1. A novel computational method to increase the cohesion and semantic coherence of a fake text to enhance believability.

2. An analysis of effects of this method on the human perception of text's believability.

The rest of the paper is organized as follows. Section 2 describes the related work on cohesion and coherence. Section 3 defines the required notations for our approach, which is described in Section 4. Section 5 describes our experimental setup, followed by result analysis in Section 6.

2 Related Work

We describe three most relevant areas in the literature to guide our methodology for improving the believability of a fake text.

2.1 Measuring Cohesion and Coherence of Text

McNamara et al. (2014) defines cohesion as "a characteristic of the text that can be computationally measured", whereas coherence is viewed as "the cognitive correlate of cohesion". Though cohesion and coherence measures have been used for evaluating student's essays (Burstein et al., 2010; Miltsakaki and Kukich, 2000), they are heavily used for evaluating automatically generated text summaries and the output of machine translation (Lapata and Barzilay, 2005). These measures describe the overlap of ideas in adjacent sentences or paragraphs. The publicly available systems of Coh-Metrix (McNamara et al., 2014) and the Tool for Automatic Analysis of Cohesion (TAACO) (Crossley et al., 2016) provide quantitative measures for cohesion, which are suitable to adapt in our research.

Lapata and Barzilay (2005) have proposed a quantitative measure of coherence based on the degree of connectivity across sentences using semantic similarity metrics. We adapt and extend their method to calculate coherence across paragraphs by computing semantic similarity between adjacent paragraphs.

2.2 Methods to Summarize and Simplify Text

Text summarization methods select salient sentences to form a short summary of the given text (Nenkova and McKeown, 2012; Erkan and Radev, 2004). Generated summaries are then smoothed to create a coherent whole out of these salient sentences (Siddharthan et al., 2011; Mani et al., 1999).

Our goal is different from text summarization, as we find salient sentences to remove them in order to reduce the knowledge that an attacker can comprehend from the document. Our approach then needs to create a coherent whole out of the remainder of the document when salient sentences are deleted. While both tasks (i.e., text summarization and believable fake document generation) find salient sentences, they focus on cohesion and coherence of different types of text units.

Another relevant research is to simplify text at the sentence and lexical levels for smoothing the generated text. Sentence level methods simplify the grammatical constructions with fewer number of modifiers (Narayan, 2014). Lexical level methods minimize the number of unique words occurring in the text (McNamara et al., 2014; Siddharthan, 2006). However, these methods are not designed to directly address the problem of linguistic inconsistency across the sentences.

2.3 Measuring Believability of Computer-generated Fake Text

An approach to measure believability of a fake text depends on the type of fake text. Fake texts can be categorized into three broad classes (Almeshekah and Spafford, 2016): manufacturing reality (curating false information from multiple documents), altering reality (modifying information in an existing document), and hiding reality (obscuring information in an existing document). A believable fake text lies at the intersection of altering reality and hiding reality. Prior literature has investigated different methods to compute the believability of such fake texts. Whitham et al. (2015) computed the difference between the k-dimensional linguistic features (e.g., word count, sentence length) of a fake text and legit text in a data repository. However this method does not evaluate the measure of believability for a human. Shabtai et al. (2016) and Bowen et al. (2009) conducted a realistic test where human readers were asked to identify the legit text from a pair of fake and legit texts. Similar to their work, we employ a believability test (more details in Section 6) to evaluate the automatically generated believable fake text.

3 Notations and Definitions

A legit text document d is used to generate a fake text document d', which is then used to generate a believable fake text document d''. Each of the documents d, d', and d'' consists of a sequence of sentences S that are grouped into K paragraphs (denoted by k_e). We define $s_i \in S$ as a salient sentence in d. The context of s_i is denoted by $c(s_i)$, where $c(s_i)$ consists of adjacent paragraphs containing $2x$ number of sentences with x number of sentences before and after s_i respectively. We define s_j to be a sentence in $c(s_i)$ that adjacently follows s_i. Document d is parsed to list the part of speech (POS) tags for each of the words in d and the list of POS tags is represented by POS_tag_list. Pronouns are recognized as p, noun phrases are recognized as n and a set of noun phrases are denoted by N. A noun phrase n follows a regular expression pattern of $Adjective * Noun+$.

Our technical approach aims to increase the cohesion and semantic coherence of a given fake text. To compute these two concepts, we use the measures of referential cohesion and semantic similarity based coherence.

Referential cohesion measures the overlap of ideas by measuring the linguistic overlap in the content words across adjacent paragraphs. We use the "adjacent_overlap_all_para" metric provided by TAACO (Crossley et al., 2016). This specific measure is defined as the number of overlapping lemma types that occur in both k_e and k_{e+1}. We compute the referential cohesion of a document d as follows:

$$Referential\_cohesion(d) = \frac{\sum_{e=1}^{count(K)-1} Referential\_cohesion(k_e, k_{e+1})}{count(K) - 1} \quad (1)$$

where k_e and k_{e+1} are adjacent paragraphs and $count(K)$ is the number of paragraphs in d.

Semantic coherence measures the overlap of ideas by assessing semantic similarity between the adjacent sentences or paragraphs. We adapt the measure proposed by Lapata and Barzilay (2005) to compute the coherence as follows:

$$Semantic\_coherence(d) = \frac{\sum_{e=1}^{count(K)-1} sim(k_e, k_{e+1})}{count(K) - 1} \quad (2)$$

where $sim(k_e, k_{e+1})$ is a measure of semantic similarity between adjacent paragraphs k_e and k_{e+1}.

We compute semantic similarity between two adjacent sentences or paragraphs using the semantic textual similarity system provided by UMBC-EBIQUITY-CORE (Han et al., 2013). This measure is based on the assumption that if two text sequences are semantically equivalent, we should be able to align their words or expressions. The alignment quality that serves as the similarity measure is computed by aligning similar words and penalizing poorly aligned words. Words or expressions are aligned using a word similarity model based on a combination of Latent Semantic Analysis (Deerwester et al., 1990) and semantic distance in the WordNet knowledge graph (Mihalcea et al., 2006).

4 Problem Statement and Solution Methodology

Problem Statement - Given an original legit text document d, generate a fake text document d' and a believable fake text document d'', where:

1. d' is fake by not containing a salient sentence s_i that is present in d,

2. d'' is believably fake by not containing a salient sentence s_i, and by following the constraints: $(Referential\_cohesion(d'') - Referential\_cohesion(d')) > 0$, and $(Semantic\_coherence(d'') - Semantic\_coherence(d')) > 0$.

Our proposed solution for believable fake text generation consists of two modules: A fake generation module and a believability module. The fake generation module consists of two operations: salient sentence identification and salient sentence deletion. The believability module consists of three operations: coreference correction, singleton entity removal, and referential cohesion improvement. We next describe each of these modules and link them to the specific functions provided in *algorithm 1*.

4.1 Fake generation module

Input: Legit text document d.
Output: Fake text document d' and deleted sentence s_i.
Objective: Generate fake text by deleting a salient sentence.
Salient sentence identification: This operation identifies the most salient sentence s_i in d using the LexRank algorithm (Erkan and Radev, 2004). LexRank computes sentence salience based on eigenvector centrality on the sentence similarity matrix, where sentence similarity is computed using idf-modified cosine similarity function.
Salient sentence deletion: This operation generates a fake text document d' by deleting s_i from the original document d.

Algorithm 1: Believability module

Input: d', s_i, POS_tag_list, θ
Output: d''

procedure BELIEVABLE_GENERATOR(d', s_i, POS_tag_list, θ)
    $temp\_d''$ = COREFERENCE_CORRECTION(d', s_j, POS_tag_list)
    $c(s_i)$ = SINGLETON_ENTITY_REMOVAL(s_i, $c(s_i)$, θ) ▷ $c(s_i)$ is extracted from $temp\_d''$
    $c(s_i)$ = REFERENTIAL_COHESION_IMPROVEMENT(s_i, $c(s_i)$, θ)
    d'' = replace $c(s_i)$ in d' with the generated $c(s_i)$
    return d''
end procedure
function COREFERENCE_CORRECTION(d', s_j, POS_tag_list)
    if s_j contains p **then** ▷ p in POS_tag_list
        compute coreference chains CC on d'
        if (p resolved to n in CC) & (s_j does not contain n) **then** ▷ n in POS_tag_list
            replace p with n
        end if
    end if
    return d'
end function
function SINGLETON_ENTITY_REMOVAL(s_i, $c(s_i)$, θ)
    Parse N_s from s_i and $N_{c(s_i)}$ from $c(s_i)$
    for each n_1 in N_s **do**
        if (n_1 not in $c(s_i)$) or (n_1 occurs more than once in $c(s_i)$) **then**
            Remove n_1 from N_s
        end if
    end for
    for each n_1 in N_s **do**
        n_2 = FIND_SEMANTICALLY_SIMILAR(n_1, $N_{c(s_i)}$, θ) ▷ n_2 in $N_{c(s_i)}$
        if REPLACEABLE(n_1, n_2) == TRUE **then**
            Replace n_1 with n_2 in $c(s_i)$
        end if
    end for
    return $c(s_i)$
end function
function REFERENTIAL_COHESION_IMPROVEMENT(s_i, $c(s_i)$, θ)
    Parse N_{before} from $S \in c(s_i)$ preceding s_i and Parse N_{after} from $S \in c(s_i)$ succeeding s_i
    for each n_1 in N_{before} **do**
        n_2 = FIND_SEMANTICALLY_SIMILAR(n_1, N_{after}, θ) ▷ n_2 in N_{after}
        if REPLACEABLE(n_1, n_2) == TRUE **then**
            Replace n_1 with n_2 in $c(s_i)$
        end if
    end for
    return $c(s_i)$
end function

4.2 Believability module

Input: Fake text document d', deleted sentence s_i, list of POS tags POS_tag_list and semantic similarity threshold between noun phrases θ.
Output: Believable fake text document d''.
Objective: Generate believable fake text by improving cohesion and coherence of text.

Next, we describe the three key sequential operations in the believability module. These operations are performed at the word level. The parts of speech of every word in d is recognized using Stanford's CoreNLP toolkit (accuracy on noun phrase tagging = 89.30%) and saved as a list - POS_tag_list.

Coreference correction (COREFERENCE_CORRECTION(d', s_j, POS_tag_list)): The purpose of this operation is to improve the ease of reading and to relate the noun phrases in $c(s_i)$. It identifies the coreference chains in the fake text using the Stanford's CoreNLP toolkit. If a pronoun p in s_j is resolved to a noun n_2, and n_2 does not occur in s_j then replace p with n_2.

Singleton entity removal (SINGLETON_ENTITY_REMOVAL(s_i, $c(s_i)$, θ): The purpose of this operation is to hide the traces of s_i in $c(s_i)$. Specifically, if there exists a noun phrase n_1 in s_i that occurs only once in $c(s_i)$ after s_i has been deleted; then, n_1 is replaced with a semantically similar noun phrase n_2 present in $c(s_i)$ (FIND_SEMANTICALLY_SIMILAR(n_1, $N_{c(s_i)}$, θ)).

Referential cohesion improvement (REFERENTIAL_COHESION_IMPROVEMENT(s_i, $c(s_i)$), θ): The purpose of this operation is to increase the cohesive relationships between the before and after parts of s_i in $c(s_i)$. First, we extract two lists of noun phrases N_{before} and N_{after} from $c(s_i)$. N_{before} is the list of noun phrases that occur in $c(s_i)$ before s_i, whereas the N_{after} is the list of noun phrases that occur in $c(s_i)$ after s_i. Second, noun phrases in N_{before} and N_{after} are compared to pair the noun phrase n_1 in N_{before} with a semantically similar noun phrase n_2 in N_{after} (FIND_SEMANTICALLY_SIMILAR(n_1, N_{after}, θ)). Finally, n_1 is replaced with n_2 in $c(s_i)$. An example of n_1 and n_2 are "methods" and "techniques" respectively.

Both singleton entity removal and referential cohesion improvement operations replace the noun phrase n_1 with another noun phrase n_2 provided n_2 is semantically similar to n_1. n_1 and n_2 are considered semantically similar if their similarity is above a threshold θ (θ=0.80 for high similarity). However, the two operations choose the noun phrases for replacement based on different criteria. Also, both these operations will replace n_1 with n_2 (REPLACEABLE(n_1, n_2)) based on the following constraints: (i) n_2 does not occur in the sentence containing n_1, (ii) n_1 and n_2 have the same plurality, (iii) n_1 and n_2 have the same number of noun terms. After n_1 is replaced by n_2, a corrective operation is performed - if n_1 is preceded by 'a' or 'an', then it is changed to suit n_2.

Next, we describe the experimental setup and the analysis of results.

5 Experimental Setup

This section presents the experimental design for testing the effectiveness of our approach. Our validation experiments are as follows:

1. **Statistical analysis** - validates the statistical significance of the improvements in cohesion and coherence of automatically generated believable fake text over the fake text.

2. **Believability test** - validates the following via human subjects: Does applying the believability module generate believable fake texts that have lower probability of being discerned than fake text?

Data: We randomly selected 25 technical articles from Communications of the ACM - a leading technical magazine. Based on the selected articles, we generated 3 sets of text documents. Each set contains 25 text documents as follows:

- *Legit text set* - First, we randomly extracted two to three consecutive paragraphs from each of the 25 original articles and created legit texts belonging to this set. The purpose of extraction is to limit the size of the documents in this set to keep it comparable to the size of context modified by the believability module.

- *Fake text set* - Next, using our fake generation module we identified the most salient sentence s_i in the original article. We also identified the context $c(s_i)$ (length of the context ($2x$) = 10) surrounding the salient sentence. Subsequently, we generated fake documents by extracting paragraphs containing $c(s_i)$ but without the salient sentence s_i.

	Fake text		Believable fake text		p-value
	Mean	SD	Mean	SD	
Cohesion	0.24	0.09	0.26	0.06	0.026
Coherence	0.37	0.10	0.40	0.09	0.013

Table 1: Comparing the change in cohesion and coherence of the fake and the believable fake texts.

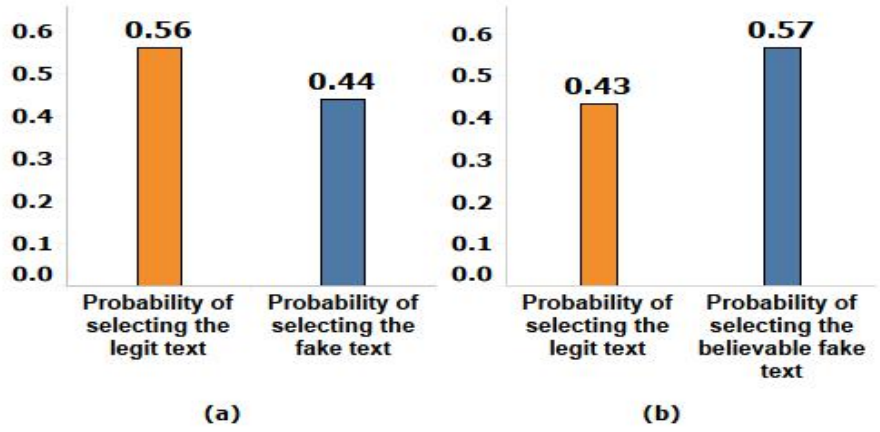

Figure 1: Aggregated analysis of 625 responses per test case of selecting the text perceived as legit - (a) given a pair of legit and fake texts (left), and (b) given a pair of legit and believable fake texts (right).

- *Believable fake text set* - Finally, we generated this set by improving the cohesion and coherence of the texts in the fake text set using our believability module.

The aforementioned method to generate sets of documents is suitable as it helps keep the legit text, fake text, and believable fake texts comparable. These texts are all extractions and modifications of consecutive paragraphs from the same original article, having the same topicality, reading level, and sharing the writing style of the same author(s).

6 Experiments and Results

This section details the experiments performed and their results.

Statistical analysis - For validating the statistical significance of the change in cohesion and coherence measures, we used the two-tailed paired t-test. We compared the 25 pairs of fake and their corresponding believable fake texts based on their cohesion and coherence measures. The results are as shown in Table 1. Looking at the p-values in the table, we can observe a statistically significant improvement in the cohesion and coherence of the text due to the operations in the believability module.

Believability test - This is a well-defined test in the domain of cyber deception that is used to test and measure the believability of a fake object. A perfectly believable fake text is one that is indistinguishable in comparison to a legit text (Bowen et al., 2009). Bowen et al. (2009) have described the procedure to conduct a believability test as follows: i) Choose two texts such that one is the believable fake text for which we wish to measure its believability and the second is chosen at random from a set of legit texts. ii) Select a human subject at random to participate in a user study. iii) Show the human subject the texts chosen in step one and ask them to decide which of the two texts is the legit text. A perfectly believable fake text is chosen with a probability greater than or equal to 50% (an outcome that would be achieved if the human subject decided completely at random).

In order to observe the change in believability due to the operations in the believability module, we conducted two types of believability tests. For the first type, we compared 25 pairs of believable fake and its corresponding legit texts derived from the same original article. We then conducted the second type of believability test where we compared 25 pairs of fake and its corresponding legit texts derived from

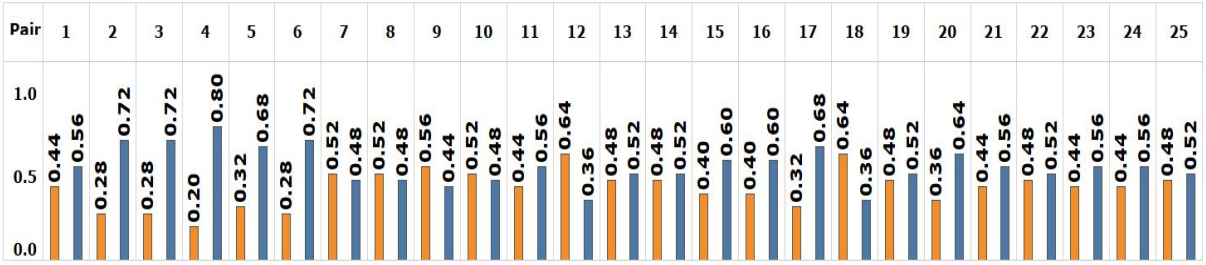

Figure 2: Distribution per test pair for 25 human subjects, where the orange bar (left) for each pair indicates the probability of identifying the legit text and blue bar (right) indicates the probability of selecting the believable fake text as the legit text.

the same original article. We did not inform the subjects about the difference in the pairs apriori. We showed each of the 50 pairs to 25 human subjects and asked them to identify the legit text. The human subjects were recruited through classes in our university and through a crowdsourcing platform (the highest trusted 'level 3' contributor set on *Figure-Eight platform*[2]). In total, we received 1250 responses for selecting the legit text in each of the 50 pairs.

We evaluated the 1250 responses using the believability test's performance metric - the probability of selecting a fake or believable fake text as the legit text. Figure 1 shows the aggregated analysis of all the 625 responses per type of believability test. Figure 1(a) shows the probability of a subject selecting the fake text as a legit text to be only 44% (p-value: 0.037, two-tailed t-test), indicating that the subjects were able to discern the legit text correctly for a statistically significant number of times. This probability indicates the likelihood of a distinguishing factor in the text that helped the subjects to identify the fake text. On the other hand, figure 1(b) shows a probability of 57% (p-value: 0.006, two-tailed t-test) for selecting a believable fake text as a legit text. This result implies that the believable fake text is truly believable for the subjects, and there may not exist a distinguishing factor that helped the subjects to recognize the believable fake text as fake.

We further performed a fine-grained analysis to validate our hypothesis that an increase in the cohesion and coherence of text would improve the believability of the text. For this analysis, we compared the individual probability of selecting a believable fake text in a believable fake-legit text pair for each of the 25 pairs. The results are as shown in figure 2. We found that 76% of the tests resulted in greater than 50% probability for a subject to identify the believable fake text as legit. These results indicate the positive effect of applying our believability module on the believability perception of fake text.

6.1 Limitations and Error Analysis

Our believability module is dependent on a semantic similarity model to provide us the similarity of noun phrases. Measuring text similarity and alignment for comparing the meaning are challenging tasks and open research questions. We chose UMBC-EBIQUITY-CORE because its similarity computation is based on leveraging both distributed semantics (Latent Semantic Analysis) and semantic networks (WordNet) for generalization. However, errors in the chosen model influences the performance of the believability module to have fewer choices when substituting similar noun phrases. Also, our approach is dependent on the POS tagger to identify noun phrases. If the tagger fails to annotate a noun or its plural form accurately, then the identified candidates for substitution would not be the complete set of nouns occurring in the document. These limitations can reduce the number of possible substitutions and therefore, limiting the possible improvements in the cohesion and coherence of the fake text.

We also conducted an error analysis on the results of the believability test to understand the characteristics of text that was not perceived as legit. In figure 2, out of the 25 pairs of believable fake-legit texts, six pairs were such that the legit text was discerned. This could be a result of pre-existing complexity in comprehending the text that was randomly chosen for generating the believable fake text. The characteristics of hard to comprehend text includes a greater presence of infrequently used words and longer

[2]https://www.figure-eight.com/

sentences. For instance, among the six pairs, we found sentences containing nearly 40 words in the chosen text. These observations motivate our future work to improve the believability by also incorporating other features of text comprehension that are beyond cohesion and coherence alone.

7 Conclusion and Future Work

We designed a novel computational linguistics method to enhance the believability of fake texts, which are used in cybersecurity solutions to deceive cyber attackers. Our methods rely on improving the linguistic consistency by increasing cohesion 1) at the sentence level via coreference correction between sentences, and 2) at the paragraph level via semantic relatedness among entities. We evaluated the outcome of our method using statistical techniques to measure the significance of improvements in the cohesion, coherence, and believability of the generated text. We found that the increase in the values of cohesion and coherence metrics for the believable fake text was statistically significant when compared with the fake text. Further, the believability test showed that the probability to distinguish a legit text from a believable fake text is lower than the probability to distinguish a legit text from a fake text. These results prove our hypothesis that the computer-generated fake text with higher cohesion and coherence leads to improvement in the believability of the text. These results further indicate the effectiveness of our method in generating believable fake text for misleading potential cyber attackers and increasing the cost of intellectual property thefts.

For the purpose of reproducibility, our dataset will be available upon request, for research purposes. Our future work will explore an extension of the newly developed methods to analyze and address the challenge of obscuring salient information at multiple locations in a given text. We will also experiment with varied types of documents by domain including non-technical documents. The application of our methods will help to create benchmark data repositories of both legit and fake text documents for cyber deception research.

8 Acknowledgement

This work was partially supported by the Office of Naval Research grants N00014-16-1-2896 and N00014-18-1-2670.

References

Mohammed H Almeshekah and Eugene H Spafford. 2016. Cyber security deception. In *Cyber Deception*, pages 23–50. Springer.

Brian M Bowen, Shlomo Hershkop, Angelos D Keromytis, and Salvatore J Stolfo. 2009. Baiting inside attackers using decoy documents. In *International Conference on Security and Privacy in Communication Systems*, pages 51–70. Springer.

Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using entity-based features to model coherence in student essays. In *Human language technologies: The 2010 annual conference of the North American chapter of the Association for Computational Linguistics*, pages 681–684. Association for Computational Linguistics.

Scott A Crossley, Kristopher Kyle, and Danielle S McNamara. 2016. The tool for the automatic analysis of text cohesion (taaco): Automatic assessment of local, global, and text cohesion. *Behavior research methods*, 48(4):1227–1237.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Gunes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Lushan Han, Abhay L Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc_ebiquity-core: semantic textual similarity systems. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 44–52. Association for Computational Linguistics.

Kristin E Heckman, Frank J Stech, Roshan K Thomas, Ben Schmoker, and Alexander W Tsow. 2015. *Cyber denial, deception and counter deception*. Springer.

Sushil Jajodia, VS Subrahmanian, Vipin Swarup, and Cliff Wang. 2016. *Cyber Deception*. Springer.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *IJCAI*, volume 5, pages 1085–1090. ACM.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically evaluating text coherence using discourse relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 997–1006. Association for Computational Linguistics.

Inderjeet Mani, Barbara Gates, and Eric Bloedorn. 1999. Improving summaries by revising them. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 558–565. Association for Computational Linguistics.

Danielle S McNamara and Walter Kintsch. 1996. Learning from texts: Effects of prior knowledge and text coherence. *Discourse processes*, 22(3):247–288.

Danielle S McNamara, Arthur C Graesser, Philip M McCarthy, and Zhiqiang Cai. 2014. *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge University Press.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st national conference on Artificial intelligence*, volume 1, pages 775–780.

Eleni Miltsakaki and Karen Kukich. 2000. Automated evaluation of coherence in student essays. In *Proceedings of LREC 2000*, pages 1–8. LREC.

Shashi Narayan. 2014. *Generating and Simplifying Sentences*. Ph.D. thesis, Universite de Lorraine.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.

Jose Otero and Walter Kintsch. 1992. Failures to detect contradictions in a text: What readers believe versus what they read. *Psychological Science*, 3(4):229–236.

Sampsa Rauti and Ville Leppanen. 2017. A survey on fake entities as a method to detect and monitor malicious activity. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pages 386–390. IEEE.

Neil C Rowe and Julian Rrushi. 2016. *Introduction to Cyberdeception*. Springer.

Asaf Shabtai, Maya Bercovitch, Lior Rokach, Ya'akov Kobi Gal, Yuval Elovici, and Erez Shmueli. 2016. Behavioral study of users when interacting with active honeytokens. *ACM Transactions on Information and System Security (TISSEC)*, 18(3):9:1–21.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2011. Information status distinctions and referring expressions: An empirical study of references to people in news summaries. *Computational Linguistics*, 37(4):811–842.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Jonathan Voris, Nathaniel Boggs, and Salvatore J Stolfo. 2012. Lost in translation: Improving decoy documents via automated translation. In *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, pages 129–133. IEEE.

Jonathan Voris, Jill Jermyn, Angelos D Keromytis, and Salvatore J Stolfo. 2013. Bait and snitch: Defending computer systems with decoys. In *Cyber Infrastructure Protection Conference*, pages 1–25. United states army college press.

Ben Whitham, Tim Turner, and Lawrie Brown. 2015. Automated processes for evaluating the realism of high-interaction honeyfiles. In *Proceedings of the 14th European Conference on Cyber Warfare and Security*, pages 307–316. Academic Conferences International Limited.

Ben Whitham. 2017. Automating the generation of enticing text content for high-interaction honeyfiles. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, pages 6069–6078. HICSS.