FORGE: A Fake Online Repository Generation Engine for Cyber Deception

Tanmoy Chakraborty, Sushil Jajodia, Jonathan Katz, Antonio Picariello, Giancarlo Sperli, and V.S. Subrahmanian

Today, major corporations and government organizations must face the reality that they will be hacked by malicious actors. In this paper, we consider the case of defending enterprises that have been successfully hacked by imposing additional *a posteriori* costs on the attacker. Our idea is simple: for every real document d, we develop methods to automatically generate a set $Fake(d)$ of fake documents that are very similar to d. The attacker who steals documents must wade through a large number of documents in detail in order to separate the real one from the fakes. Our FORGE system focuses on technical documents (e.g. engineering/design documents) and involves three major innovations. First, we represent the semantic content of documents via multi-layer graphs (MLGs). Second, we propose a novel concept of "meta-centrality" for multi-layer graphs. A meta-centrality (MC) measure takes a classical centrality measure (for ordinary graphs, not MLGs) as input, and generalizes it to MLGs. The idea is to generate fake documents by replacing concepts on the basis of meta-centrality with related concepts according to an ontology. Our third innovation is to show that the problem of generating the set $Fake(d)$ of fakes can be viewed as an optimization problem. We prove that this problem is NP-complete and then develop efficient heuristics to solve it in practice. We ran detailed experiments on two datasets: one a panel of 20 human subjects, another with a panel of 10. Our results show that FORGE generates highly believable fakes.

***Index Terms*—Fake documents, online repository, cyber deception.**

I. Introduction

According to the 2017 Verizon Data Breach Investigations Report, 75% of data breaches were perpetrated by external actors. 18% of all data breaches were carried out by state actors, while 51% were orchestrated by criminal groups. Moreover, in some sectors such as education, only 18% of vulnerabilities for which patches were available were in fact patched after more than 12 weeks. Even the finance industry only fixed 33% of their vulnerabilities after 12 weeks. Most industry sectors had more than half of their vulnerabilities unpatched after 2 weeks. These findings by Verizon make it clear that even in industry sectors such as IT and manufacturing, an attacker can use existing exploits for vulnerabilities to target enterprises even when patches have been released because vulnerabilities can stay unpatched for weeks. Theft of intellectual property (IP) and other protected data from there is a simple next step.

In this paper, we propose a simple, yet novel method to increase costs on attackers who wish to steal documents from an organization. For each document d in a given repository $\mathcal{R}$, the proposed FORGE system augments $\mathcal{R}$ with many near identical, but fake, copies of each file d. An intruder who has penetrated the enterprise network would access the augmented set $\mathcal{R}^\star$ of files and would face at least one of two problems. First, he may not be aware that the system has many fake files. Because external connections and exfiltration of data is monitored closely by most organizations with sensitive information, there is a high probability that the attacker will steal an incorrect file. If such files include, for instance, complex designs for an aircraft or a submarine, the attacker would incur actual costs (dollars) and time delays as he tries to execute the design, only to find months later that it doesn't work. Second, if the attacker expects the file system to contain fake documents, he could either (i) spend more time within the system picking and trying to choose the right document, thus at the very least increasing risk of discovery, or (ii) he could exfiltrate many documents in the hope of going through them at leisure within his network - but this too increases risk of discovery. Thus, in all cases, creating a repository of both real and fake documents increases costs for the attacker. However, for a fake version d' of an original document d to be successful in deceiving the attacker, it must be believable.

In order to achieve these tasks, we start by showing that any given document can be represented as a multi-layer graph (MLG for short) [1]. We then propose two highly novel contributions by leveraging this MLG representation.

1) First, we propose the novel concept of *Meta-Centrality (MC)* to measure the importance of a concept in the MLG representation of a document. MC takes as input, not only a document, but also a standard centrality measure (e.g., betweenness or eigenvector centrality) for ordinary (single) layer graphs, and produces a measure of the importance of a concept to a document by building on top of that centrality. To the best of our knowledge, this is the first notion of meta-centrality proposed in the literature, as well as the first notion of centrality for MLGs. The idea is that an ontology can be used to replace certain concepts (within some MC range) in a document by others that are plausible replacements.
2) Second, we show that the problem of generating not one, but a set of fake documents from a single original

T. Chakraborty is with Indraprastha Institute of Information Technology, Delhi (IIIT-D), India, e-mail: tanmoy@iiitd.ac.in

S. Jajodia is with George Mason University, USA, e-mail: jajodia@gmu.edu

J. Katz is with University of Maryland, College Park, USA, e-mail: jkatz@cs.umd.edu

A. Picariello and G. Sperli are with University of Naples "Federico II", e-mail: {antonio.picariello,giancarlo.sperli}@unina.it

V.S. Subrahmanian is with Dartmouth College, USA, e-mail: vs@dartmouth.edu

document can be solved via *Integer Linear Programming*. We show that the problem of generating a fake repository $\mathcal{R}^\star$ from $\mathcal{R}$ is NP-complete and propose a heuristic algorithm for this purpose.

We have developed a prototype implementation of the FORGE system that works on patent documents from the agricultural/chemical industry[1]. We ran two experiments. First, using a panel of 20 engineering MS students, we experimentally show that FORGE generates a repository that successfully cloaks the identity of the real documents in the repository with at least 94% accuracy. Second, with a panel of 10 MS computer science students and a collection of computer science documents, we show that our system cloaks the identity of the real documents with over 92% accuracy.

In most cases, human subjects pick the wrong documents and hence, FORGE is able to deceive adversaries. FORGE also proposes a simple cryptographic approach based on method authentication codes so that legitimate users can distinguish between a real document and a fake version of it.

II. Related Work

Deception is not new [9], [10]. For instance, the recording industry combated piracy by flooding the Internet with fake MP3 songs that blast horrendous music when played [11]. [2] suggested generating decoy "honey files" that attempt to lure attackers in order to improve intrusion detection. In such honey schemes, system security officers are alerted if a honey file (e.g. "password.txt") is accessed, The D^3 system [3] builds attractively named decoys which contain several embedded monitors to detect exfiltration – for instance, they embed a 'beacon' that reaches back to a command and control server when an exfiltrated document is opened by a malicious hacker. [4] generates believable Java source code using code obfuscation techniques to translate original software using various translation methods. Using a document similarity measure, they showed that the generated bogus software is very different from the original one while maintaining a similar level of complexity. In order to deal with fraud, [5] proposed a method to detect stylistic deception in written documents by exploiting linguistic features to distinguish regular documents from deceptive documents. Voris et al. [12] proposed an automated foreign language translation system that generates foreign language decoy text and sprinkles it with untranslatable, but enticing nouns such as company names, hot topics, and apparent login information. They argued that augmenting a foreign language that is rarely used in a given organization into the document offers a clear signal to legitimate users that it is fake, whereas an attacker still needs to exfiltrate the document in order to translate it. [6] proposed using a fake "Canary File" in order to detect unauthorized data access, copying or modification. The Canary File acts as a hidden watermark for a file directory containing critical documents. Later, [6] introduced a set of requirements for fake file content generation: they should be enticing, realistic, and cause minimal disruption [13]. [14] further states that the key challenge of any fake document generation system is the burden of generating deceptive content which appears believable to the attackers. [7] additionally states that it usually takes several months of research to produce a realistic decoy by examining all the topics that can be manipulated to generate fake documents. A recent solution [8] requires a separate new module to be deployed for each topic, which is cumbersome and time consuming. [3] argues that balancing the amount of content disruption and number of fake documents generated is a critical aspect of any practical fake file generation system.

The major limitations of existing approaches mentioned above are as follows:

- *Understanding replaceable topic/concepts:* Although [8] suggested designing different modules for different topics to be replaced, they do not specify how to choose topics in a document that should be replaced. Our FORGE system proposes a novel notion of meta-centrality (MC), together with many MC measures and detailed experimental results on how best to make this choice.
- *Handling technical articles:* Existing decoy document generation systems [3], [7], [8] attempt to change biographic information (such as name, credit card details, SSN number, ATP pin code, address, telephone number) present in an original document to generate believable fake documents. However, the needs of technical documents such as intellectual property, trade secrets, and internal engineering designs that a company may wish to process have very different requirements. FORGE focuses on technical documents such as patents, chemical composition/process documents, engineering designs, and so forth where there is no biographic information;
- *Handling balance:* The existing literature does not show how to control the balance between the amount of fake content and the number of fake documents generated [3]. FORGE handles balance by mapping the fake document generation problem to a combinatorial optimization problem which provides a mechanism to generate a set of believable fake documents within a certain enterprise budget.
- *Human in the loop feedback:* Most existing techniques do not allow the security officer to customize the system if the fake documents generated are not satisfactory. Ideally, the system security officer should be able to inspect automatically generated fake documents and make necessary changes. FORGE supports this by providing system security officers with the opportunity to accept/reject the suggested replacing/replaced concepts or add their own choices into the system. Our online learning module retrains the system after such user feedback is provided.
- *Human evaluation:* Most existing literature lacks an appropriate subjective evaluation to justify the quality of the generated fake documents. In contrast, we performed an evaluation via a panel of 20 human subjects.

Table I compares FORGE with past work and shows that FORGE addresses all the limitations mentioned above. To the best of our knowledge, FORGE is the first fake document

[1] While our method is practically demonstrated on intellectual property only (as opposed to say, social security or bank account numbers), we believe the same principles will apply to those domains as well.

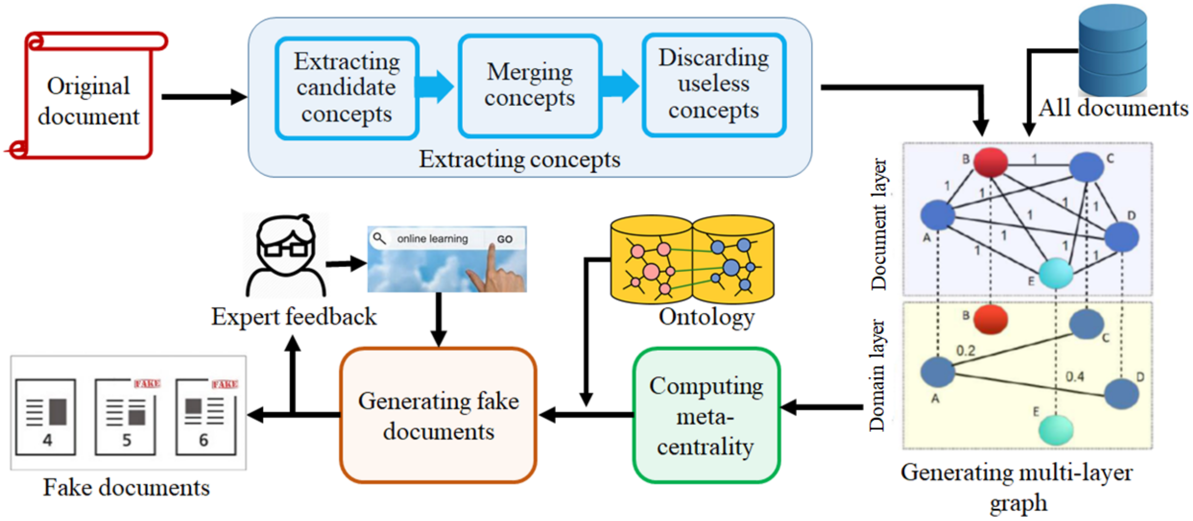


Fig. 1: The flowchart of the proposed framework FORGE.

TABLE I: Comparison of FORGE with existing systems.

Existing System	Replaceable topic	Technical docs	Balance factor	Deployable system	Human in loop system	Human evaluation
Yuil et al. [2]	×	×	×	✓	×	×
Bowen et al. [3]	×	×	×	✓	✓	×
Park and Stolfo [4]	×	×	×	✓	×	×
Afroz et al. [5]	×	✓	×	✓	×	×
Whitham [6]	×	×	×	✓	✓	×
White and Thompson [7]	×	×	×	✓	✓	×
Wang et al. [8]	×	×	×	✓	✓	×
FORGE	✓	✓	✓	✓	✓	✓

TABLE II: Notation used in this paper and their descriptions.

Notation	Description
d	An original document
d'	$d' \in Fake(d)$, a fake version of d
c	A concept
$\mathcal{R}$	A repository of real documents
$\mathcal{R}^\star$	A repository containing both real and fake documents
G_s^d	$G_s^d = (V_s^d, E_s^d, \omega_s^d)$, the document layer of MLG
G_c^d	$G_c^d = (V_c^d, E_c^d, \omega_c^d)$, the domain layer of MLG
A^c (*resp.* A^s)	Adjacency matrix corresponding to the document (*resp* domain) layer
$CM_s(v)$	A classical single-layer centrality measure of v applied on document layer G_s^d
$MC(c)$	Meta-centrality of concept c
$MCR(c)$	Meta-centrality based rank of a concept c
ac_i	Alternative candidate concept of concept c_i
$Cost(.,.)$	Concept substitution cost
$dist_{\mathcal{O}}(c_i, c_j)$	Distance of concepts c_i and c_i in ontology $\mathcal{O}$
C	Set of concepts we want to substitute

generation framework that is specifically designed to protect technical documents by identifying central concepts of a document and replacing them with believable fake concepts.

III. Proposed Methodology

We have developed a mathematical theory and a system called FORGE (short for) **F**ake **O**nline **R**epository **G**eneration **E**ngine. FORGE starts by extracting concepts from a given document and builds a multi-layer graph (MLG) by considering how extracted concepts are related to each other inside the document as well as inside the whole repository. Following this, it measures the importance of a concept by a novel family of "Meta-Centrality" (MC) metrics which build upon classical graph-based centrality measures. Finally, FORGE maps the fake document generation problem to a combinatorial optimization problem. We also show that the decision variant of this optimization problem is NP-complete, and therefore use a heuristic algorithm to come up with a practically usable version. A flowchart of the entire framework is shown in Figure 1. Important notations used in this paper are summarized in Table II.

A. *Extracting Concepts*

We extract the concepts in a document[2] d using standard methods via three "shallow NLP" steps:

1) *Extracting candidate concepts*: Given a document d, we first extract equations and formulas from the document using the html tags present in the document. We then use the Stanford NLP parser[3] to identify and extract all noun phrases of size 5 or less from the document.
2) *Discarding redundant concepts*: Once we identify a set of candidate concepts, we discard redundant concepts. For instance, if one concept is a proper subset of another

[2] The FORGE implementation works on HTML documents. There is no loss of generality in this assumption as there are many free, off the shelf converters that transform PDF, DOC, and DOCX files into HTML.

[3] http://nlp.stanford.edu/software/

concept, we discard the former (smaller) one and only consider the latter (larger) cone.

3) *Discarding useless concepts*: We define a set of rules to discard concepts that are not related to the domain of the repository. For instance, if we seek to protect intellectual property of an aerospace company, we would remove documents related to the company's employee health plans.

Example III.1 (Extracting Concepts)**.** *Consider the following text snippet from a patent [15]:*

> A surfactant composition for agricultural chemicals containing fatty acid polyoxyalkylene alkyl ether expressed by the following formula (I):
>
> $$R^1CO(EO)_{rn}(PO)_nOR^2 \ \ (I)$$
>
> wherein the fatty acid polyoxyalkylene alkyl ether has a narrow ratio of 55% by mass or more, where the narrow ratio is expressed by the following formula:
>
> $$Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i \ \ (A)$$

1) *Extracting candidate concepts:* FORGE *extracts the following set of concepts.* {Surfactant composition, Agricultural Chemicals, fatty acid polyoxyalkylene alkyl ether, Narrow ratio, $Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i$, R^1CO, $R^1CO(EO)_{rn}(PO)_nOR^2$, following formula, mass or more}.
2) *Discarding redundant concepts: Note that some concepts in the above set are part of other concepts. For instance,* R^1CO *is a part of* $R^1CO(EO)_{rn}(PO)_nOR^2$*;* Narrow Ratio *is included in* $Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i$*. After the merging step,* FORGE *returns the following set of concepts* {Surfactant composition, Agricultural Chemicals, fatty acid polyoxyalkylene alkyl ether, $Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i$, $R^1CO(EO)_{rn}(PO)_nOR^2$, following formula, mass or more}.
3) *Discarding useless concepts:* FORGE*'s filtering rules discard useless words such as "following formula", "mass or more", yielding the following set:* {Surfactant composition, Agricultural Chemicals, fatty acid polyoxyalkylene alkyl ether, Narrow ratio, $Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i$, R^1CO, $R^1CO(EO)_{rn}(PO)_nOR^2$}.

We would like to emphasize that we are not claiming that our concept extraction methods are novel. In fact, there is a very rich body of work on concept extraction from documents [16], [17] and FORGE merely leverages this work.

B. From Documents to Multi-layer Graphs

Once we collect the filtered list of concepts from a document, we build a multi-layer graph (MLG) [18], consisting of two layers – a *document layer* and a *domain layer*. The multi-layer graph associated with a document captures both the relationships between concepts inside the document as well as across the domain related to the document.

A vertex in the multi-layer graph of a document is a concept and an edge connecting two vertices represents the similarity between the two concepts. In this section, we describe how we associate an MLG with a graph.

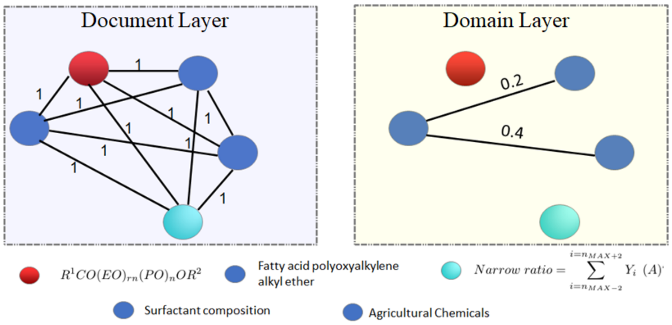

Document Layer
1
1
1
1
1
1
1
1
1
1
Domain Layer
0.2
0.4
$R^1CO(EO)_{rn}(PO)_nOR^2$
Fatty acid polyoxyalkylene alkyl ether
$Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i \ (A)$
Surfactant composition
Agricultural Chemicals

Fig. 2: Document and domain layers associated with Example III.1 using a sliding window of size $k = 5$. This leads to a completely connected document layer. On the other hand, the domain layer is disconnected.

Definition III.1 (Document Layer)**.** *Given a document* d*, the* document layer *is a graph* $G_s^d = (V_s^d, E_s^d, \omega_s^d)$ *whose vertices are the concepts extracted from* d*. There is an edge between two concepts if they appear within a sliding window of size* k *or less (i.e., the two concepts are separated by at most* $k-1$ *words) inside* d*.* $\omega_s^d : C \times C \rightarrow \mathbf{N}$ *(where* C *is the set of concepts) is a weight function such that* $\omega_s^d(c_i, c_j)$ *is the number of times both* c_i *and* c_j *appear within the sliding window in document* d*.*

Note that $\omega_s^d(c_i, c_j) = 0$ if and only if c_i, c_j never appear within a sliding window of size k which is equivalent to saying that there is no edge linking these two concepts in the document layer. The document layer captures the idea that if two concepts appear together multiple times inside a document, they are likely to be linked.

Example III.2 (Document Layer)**.** *Figure 2 shows the document layer associated with the concepts extracted in Example III.1. We use an example with a sliding window of size* $k = 5$*.*

The document layer forms the first layer of the MLG associated with a document (to be formally defined in Definition III.4). In order to construct the second layer, we introduce a notion of 'domain' of a concept.

Terminology III.1 (Context of a Concept)**.** *Given a repository* $\mathcal{R}$ *of documents and a concept* c*, the* domain $\Gamma(c)$ *of* c *consists of the set of all words appearing* k *positions before or after* c *in any document in* $\mathcal{R}$*.*

The underlying idea is that "a concept is characterized by the company it keeps" [19]. This is also known as "distributional semantics" in NLP [19].

The second layer of the MLG associated with a document d w.r.t. a repository $\mathcal{R}$ measures the similarity between concepts by considering the overlap between their contexts. If two concepts in a document "keep similar" company in $\mathcal{R}$, i.e., their contexts are similar across the repository as a whole, then they are considered to be linked. We use Jaccard coefficients to measure similarity.

Definition III.2 (Similarity between two concepts)**.** *Given*

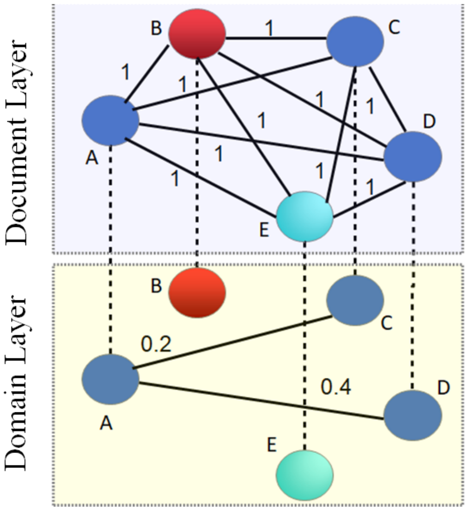


Fig. 3: A schematic diagram of a multi-layer graph $G^d = (V^d, E^d)$ which is composed of two layers – document layer (G^d_s) and domain layer (G^d_k). The dotted lines are the inter-later edges, connecting two different instances of a same concept. In Section III-C, we will show how to computer meta-centrality of all the vertices in G^d.

two concepts c_i and c_j and their contexts $\Gamma(c_i)$ and $\Gamma(c_j)$ respectively, the similarity between them is measured by their Jaccard Co-efficient JC: $JC(c_i, c_j) = \frac{|\Gamma(c_i) \cap \Gamma(c_j)|}{|\Gamma(c_i) \cup \Gamma(c_j)|}$.

Definition III.3 (Domain Layer)**.** *Given a document d and a repository $\mathcal{R}$, the* domain layer *is a graph $G^d_c = (V^d_c, E^d_c, \omega^d_c)$ where V^d_c is the set of concepts C extracted from d. The set E^d_c of edges is the set $\{(c_i, c_j) \mid JC(c_i, c_j) > 0\}$. The weight function ω^d_c is defined as $\omega^d_c(c_i, c_j) = JC(c_i, c_j)$.*

The idea behind the domain layer stems from a popular concept called "distributional semantics" [20] in linguistics which suggests that the words that are used and occur in the same contexts tend to have similar meanings. Essentially, we compare two sets of words $\Gamma(c_i)$ and $\Gamma(c_j)$. Other similar measures such as cosine similarity, Euclidean distance are not applicable here.

Example III.3 (Domain Layer)**.** *Figure 2 shows a domain layer in connection with the concepts extracted in Example III.1. We use the entire document repository to identify similarities between the main concepts.*

We are now ready to define the multi-layer graph (MLG) associated with a document d and a repository $\mathcal{R}$.

Definition III.4 (Multi-layer Graph)**.** *Given a document d, a repository $\mathcal{R}$, and the two layers G^d_s and G^d_c, the associated* multi-layer graph *is the pair $G^d = (V^d, E^d)$ where $V^d = \{V^d_s \cup V^d_c\}$, $E^d = \{E^d_s \cup E^d_c \cup E_{sc}\}$ and E_{sc} is the set of inter-layer links connecting the same concepts in both the layers. Note that the vertex set is the same in both layers (i.e., $V^d_s = V^d_k$).*

A schematic diagram of the multi-layer graph generated from our example is shown in Figure 3. In short, the multi-layer graph captures how the concepts are similar to each other within a given document (document layer) and across all other documents in a repository (domain layer).[4]

C. Meta-Centrality

We now propose the notion of *meta-centrality* in which we define the centrality of nodes within an MLG by building upon centrality measures that have been tried and tested for ordinary (single layer) graphs. The notion of meta-centrality is a recursive definition.

Definition III.5 (Meta-Centrality)**.** *Given a classical centrality measure CM for single layer graphs, a document d, and its multi-layer graph $G^d = (V^d, E^d)$ consisting of two layers G^d_s and G^d_c (where A^s and A^c correspond to the adjacency matrices of G^d_s and G^d_c respectively), the meta-centrality of a concept c_i (corresponding to a vertex $v_i \in V^d$) is a function MC that maps vertices (corresponding to concepts) to $\mathbf{R}$ as follows:*

$$MC(c_i) = \beta \sum_j A^c_{ji} \frac{\frac{CM_s(v_j)}{\Sigma_{v \in V^d_s} CM_s(v)} + MC(c_j)}{max(1, d^c_j)} + (1-\beta) \frac{CM_s(v_i)}{N\langle X^s \rangle} \tag{1}$$

where:

- *$\beta \in [0, 1]$ is a parameter controlling the weight of two components,*
- *CM_s is a classical centrality measure applied to the document layer G^d_s,*
- *d^c_j is the degree of vertex v_j in the domain layer G^d_c,*
- *$N = |V^d_s| = |V^d_c|$,*
- *$\langle X^s \rangle$ is the mean of graph-based centrality values of vertices computed in the document layer, i.e., $\langle X^s \rangle = \frac{1}{N} \sum_{v_i \in V^d_s} CM_s(v_i)$.*

The definition of meta-centrality takes a standard centrality measure and applies it to the document layer. When computing meta-centrality of c_i, two terms need to be summed. The second term is merely the normalized centrality of c_i in the document graph, multiplied by a constant $(1 - \beta)$. The first term is a PageRank style formula. It looks at all concepts c_j that are connected to c_i. For each such concept, it computes a normalized centrality of c_j in the domain layer and then adds this to the meta-centrality of c_j. This is then multiplied by the weight A^c_{ji} of the edge between c_i, c_j in the domain layer and equally allocated (hence the division) across all neighbors of c_j. The result is weighted by β. Note that when $\beta = 0$, $MC(c_i)$ depends solely on the second term and when it is 1, it depends solely on the first term. Thus, β is a parameter. Later in the paper, we will experiment with the two major parameters in this definition: β and CM.

Equation 1 builds on the PageRank formula. However in this case, we first compute centrality values of vertices in the document layer and then adjust the score based on the connectivity in the domain layer. Moreover, we place greater

[4] We note that an MLG such as the one proposed here can also be represented as a graph where an edge can have more than one label. For our purposes, either of these two representations will do — we pick the MLG representation because our notion of meta-centrality is applicable to MLGs in general.

emphasis on the document layer in Equation 1 because the document layer captures the relationship between concepts specific to a document, whereas the domain layer captures the relationship across concepts in the entire domain (which is same for all the other documents in that domain). Later (Section IV), we will show results corresponding to different parameter settings of meta-centrality.

We can now compute the meta-centrality of a concept w.r.t. the MLG associated with a document and a repository in the usual way used by PageRank. We start by setting all the concepts to have equal value and apply the formula in the definition of meta-centrality to each concept in order to get a new set of values. This process is repeated either till convergence occurs or till a certain fixed number of iterations has been performed. Convergence is usually defined to occur when the current iteration $t+1$ is not able to significantly alter the meta-centrality of the concepts obtained from iteration t. Then we expect that the algorithm reaches a "steady state." In mathematical terms, the algorithm converges when $|MC(c_i, t+1) - MC(c_i, t)| < \epsilon$, for some ϵ. ϵ is commonly set to $1.0e-4$ in implementations of PageRank [21] and we do the same.

In this paper, we use the following four classical centrality measures [22] CM to compute MC.

1) *Degree Centrality* (DC): Given a graph $G = (V, E)$, degree centrality $C_D(.)$ of a node v is measured by the number of links incident upon the node, normalized by the number of edges, i.e., $C_D(v) = \frac{degree(v)}{|E|}$.
2) *Betweenness Centrality* (BC): Given a graph $G = (V, E)$, betweenness centrality $C_B(.)$ of a node v measures the probability that a node appears on the shortest path between two other nodes, i.e., $C_B(v) = \sum_{s \neq v \neq t \in V)} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} is total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.
3) *Closeness Centrality* (CC): Given a graph $G = (V, E)$, closeness centrality $C_C(.)$ of a node v is the reciprocal of the sum of the shortest path from node v to every other node in the graph, i.e., $C_C(v) = \sum_{v \in V \& v \neq u} \frac{1}{dist(u,v)}$, where $dist(u, v)$ is the shortest-path distance between u and v.
4) *PageRank* (PR): Given a graph $G = (V, E)$, PageRank $C_{PR}(.)$ of a node v is measured by $C_{PR}(v) = \alpha \sum_{u \in N(v)} \frac{C_{PR}(u)}{|N(u)|} + \frac{1-\alpha}{|V|}$, where $N(v)$ is the set of nodes directly connected with v, and α is a damping factor (usually set to 0.85).

Thus meta-centrality is not a single measure, but a family of measures depending on which classical centrality measure is used and what value of β is used.

Example III.4 (Meta-Centrality)**.** *We now show how to compute meta-centrality of the vertex A in the multi-layer graph of Figure 3. If we consider degree centrality as the single-layer graph centrality measure, the centrality value of A in the document layer G_s^d, denoted by $CM_s(A)$, is $\frac{2}{5}$. Similarly, we compute the degree centrality value of other vertices in G_s^d, and measure the mean centrality score $\langle X^s \rangle = \frac{1}{5} \sum_{v \in \{A,B,C,D,E\}} CM_s(v) = \frac{2}{5}$. The degree of A in the context layer G_c^d, denoted by d_c^A, is 2, and $N = 5$. Considering $\beta = 0.85$ and initializing $\forall v \in V_d^c : MC(v) = \frac{1}{N}$, we recursively compute the meta-centrality values for all the vertices (concepts), until it converges (i.e., there is not significant change in the value of meta-centrality of concepts). Finally, we obtain $MC(A) = 0.64$. The meta-centrality value for other vertices of Figure 3 using different graph-based centrality measures are shown in Table III — though it is unusual, in this example, the meta-centrality based on both closeness centrality and PageRank turn out to be the same.*

TABLE III: Meta-centrality values of vertices in Figure 3 using different single-layer graph centrality measures.

	Degree	**Betweenness**	**Closeness**	**PageRank**
A	0.64	0.10	0.64	0.64
B	0.03	0.0	0.03	0.03
C	0.31	0.02	0.31	0.31
D	0.59	0.03	0.59	0.59
E	0.03	0.0	0.03	0.03

Once we obtain the meta-centrality value of each concept, we rank them in decreasing order. *In the rest of the paper, $MCR_d(c)$ denotes the rank of concept c based on the decreasing order of meta-centrality value in document d.*

Example III.5 (Meta-Centrality Ranking)**.** *For Figure 3, based on the meta-centrality value of nodes mentioned in Example III.4 by considering degree centrality in the document layer, the meta-centrality ranking of nodes (concepts) is given by: $MCR(A) = 1$, $MCR(B) = 4$, $MCR(C) = 3$, $MCR(D) = 2$ and $MCR(E) = 4$.*[5]

Before concluding this section, we note that many notions of centrality exist in the information retrieval literature. [23] defined *semantic centrality* as the power of controlling semantic information flow on a social network. Exploiting this definition further, Leprovost et al. [24] provided another definition of *semantic centrality* that takes into account both semantics and communication timestamps together. Traub et al. [25] defined *semantic centrality* as a measure how central a text is among a collection of texts, in terms of its semantic overlap or similarity with all other texts.

However, our notion of *meta-centrality* differs in many ways. First, it is defined on concepts while the above definitions are not. Second, it is defined on a multi-layer graph , not an ordinary graph. This captures two types of similarity among concepts – 'local similarity' within a document and 'global similarity' within a domain (across different documents). Third, it can take any classical notion of centrality as an input parameter - thus meta-centrality is not a single centrality measure but a whole family of centrality measures that extend classical centrality measures.

D. Generating Fake Documents

Once we obtain the meta-centrality rank of each concept in document d, the next task is to generate fake versions of d by

[5] Note that we use 'dense ranking', i.e., items that have equal meta-centrality receive the same rank.

replacing some concepts with “similar concepts” in such a way that the fake documents appear “believable” to the attackers. However, it is unclear which concepts should be replaced. Although meta-centrality provides a rank of concepts in a document, we do not know a-priori if replacing top-ranked concepts would generate a believable document. Moreover, replacement of a concept by another concept incurs a specific cost in terms of believability, depending both on what is replaced and what the replacement is.

In order to explain the fake document generation process, we first present some important definitions and terminologies.

Terminology III.2 (Set of Fake Documents)**.** *Given an original document d, we define a* set $Fake(d)$ of fake documents *as an ordered list of documents $\{d'_1, d'_2, \cdots\}$ where each fake document d'_i is generated by replacing selected concepts in d by other alternative concepts.*

The questions that must be addressed are the following – (i) which candidate concepts can replace a given concept? (ii) which concepts should be replaced? (iii) how many concepts should we replace? The rest of the section answers these questions separately.

1) Alternative Candidate Concept Generation

In order to identify the candidate concepts to replace c, we use a domain-specific ontology $\mathcal{O}$. An ontology allows us to model knowledge about a domain by identifying specific classes of concepts, defining each instance of concepts through properties or attributes, and defining different types of relationships between concepts.

Definition III.6 (Ontology)**.** *An ontology O is a pair (D, R) where D is a set called the universe of discourse and R is a set of binary relations on D [26].*

There is now an extensive range of ontologies that are available on a domain by domain basis — for instance, http://info.slis.indiana.edu/~dingying/Teaching/S604/OntologyList.html presents links to an exhaustive set of ontologies for a variety of domains including customer complaints, music, images, biology, health care, and more. Figure 4 shows a small sample chemical ontology produced verbatim from the literature[6].

Terminology III.3 (Alternative Candidate Concepts)**.** *Given an ontology $\mathcal{O} = (D, R)$ and a concept c_i, the set of* alternative candidate concepts $\{ac^i_1, ac^i_2, \cdots\} \subseteq D$ *consists of members of D whose distance from c_i is less than or equal to a given threshold.*[7] *In some cases, a concept extracted by* FORGE *may consist of many words and the concept by itself may not match any term in the ontology. In this case, we search for the constituent words of C_i individually in $\mathcal{D}$ and replace them with the possible alternatives as defined above.*

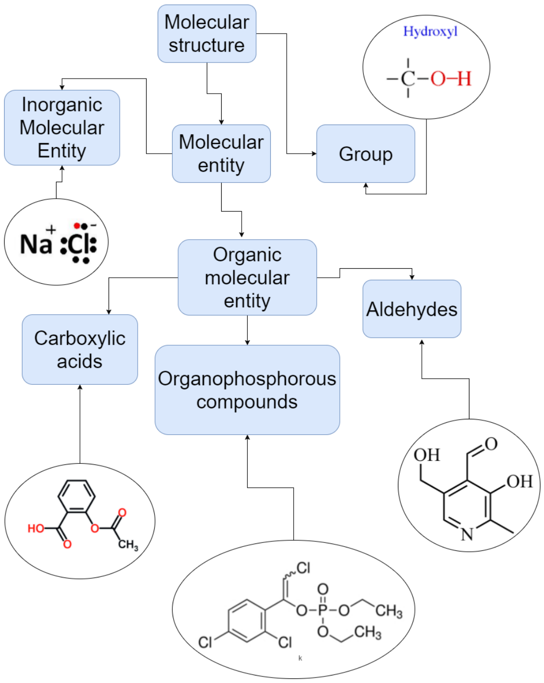


Fig. 4: Example of a chemical ontology.

For example, suppose we consider the concept “sodium carbonate” and the chemical periodic table is the ontology. Since lithium, potassium etc. are in the same group as sodium, we obtain several alternative candidates – lithium carbonate, potassium carbonate etc. Some of these might not constitute a valid substitution — for instance, aluminum carbonate may not be a compound that exists in practice. We verify the validity of a candidate by searching for it in a collection of documents (e.g., in an archive of papers published in a leading chemistry journal). However, substituting a concept C_i with an alternative concept aC^i_j involves a cost in terms of believability because the substituted concept may or may not make sense to a domain expertise. Below we define the cost of substituting a concept with another concept.

Definition III.7 (Substitution Cost)**.** *Given an ontology $\mathcal{O}$, two concepts (c_i and ac_j) and a distance function $dist : C \times C \to R$, we define the cost of substituting c_i with ac_j as: $Cost(c_i, ac_j) = dist_{\mathcal{O}}(c_i, ac_j) = \sum_k dist_{\mathcal{O}}(c^k_i, ac^k_j)$, where c^k_i and ac^k_j are the words in c_i and ac_j respectively (assuming that a concept is composed of multiple words), and $dist(.,.)$ is an ontological distance between two concepts (e.g., shortest path). Note that the length of c_i and ac_j is the same because ac_j is generated after component-wise substitution of c_i.*

In short, we take a document d, and generate a set of fake documents by replacing some concepts in d by other concepts drawn from the ontology. By requiring that the total substitution cost is below a given threshold, we ensure that concepts in the real document are replaced by concepts “nearby” in the ontology, thus enhancing believability.

2) Substituted Concepts Identification

In order to answer two other questions – which and how many concepts should be substituted, we map the fake

[6] https://image.slidesharecdn.com/chemicalontologygcc2010-101108065902-phpapp02/95/chemical-ontologies-what-are-they-what-are-they-for-and-what-are-the-challenges-8-638.jpg?cb=1422620039

[7] Because an ontology includes in its definition, a set of binary relations, we can easily represent it as an undirected graph and compute the distance between two concepts to be the shortest path distance in this graph. Other notions of distance between concepts can also be used and later in the paper, we will also introduce an entropy based distance – but for the sake of simplicity, we proceed with graph based distance for now.

document generation problem to an optimization problem constrained on the overall budget to generate fake documents and the number of candidate concepts to be replaced. Because we do not know in advance whether highly central concepts in the document should be replaced or medium-ranked ones, or low-ranked ones, we introduce two parameters, n_1, n_2 where $0 \leq n_1 \leq n_2 \leq 1$ which regulate whether highly ranked, medium-ranked, or low-ranked concepts should be selected for substitution. Later, we will run experiments varying the values of n_1, n_2 in order to find the ones that achieve the best deception results. We systematically select these concepts from different positions of the rank-list obtained from the meta-centrality values. The proposed constrained optimization function is as follows:

$$\begin{aligned}
&\hat{C} \leftarrow \underset{C}{\text{argmin}} \quad \sum_{i=1}^{|C|} MCR_d(c_i) \\
&\text{subject to} \\
&(i)\ n_1C \leq MCR_d(c_i) \leq n_2C : \begin{cases} n_1 = 0,\ 0 < n_2 \leq 1 : \text{ Upper} \\ 0 \leq n_1 < n_2 \leq 1 : \text{ Middle} \\ 0 \leq n_1 < 1,\ n_2 = 1 : \text{ Lower} \end{cases} \\
&(ii)\ n \geq 1; \\
&(iii) \sum_{k=1}^{|C_j|} Cost(c_k, ac_k) X_{kj} \leq \mathcal{B}_j;\ \forall j \\
&(iv) |C_j| \leq |C|; \\
&(v) \sum_{j=1}^{n} X_{kj} = 1; \quad \forall k : \\
&(vi)\ X_{kj} \in \{0, 1\}; \quad \forall k :
\end{aligned} \tag{2}$$

Given an original document d and a budget $\mathcal{B}$, we generate a set of n fake documents $Fake(d) = \{d'_1, d'_2, \cdots, d'_n\}$ by choosing (and substituting) C concepts in such a way that the sum of meta-centrality ranking MCR of the chosen concepts in d is minimized (as low ranks are good) subject to the requirement that meta-centrality ranks fall within a given interval (see constraint (i) in Equation 2). Note that we choose to minimize the sum of the meta-centrality *ranks* as opposed to the meta-centrality scores because the ranks place a relative rank order on the concepts. In addition, a rnk such as 1 is better than a rank such as 5 which is why we minimize the sums. That said, we do not know a priori which concepts should be replaced - highly central ones? moderately central ones? or low centrality concepts? One may argue that substituting the top-ranked concepts in the original document might make it easy for a malicious user to infer that a document is fake. We investigate all three options experimentally later in the paper (see Section IV). In order to choose an efficient strategy we introduce constraint (i) into our problem. In particular, we consider the same objective function mentioned in (2) and impose three additional constraints separately to generate three versions of the optimization problem:

1) **Upper:** $n_1C \leq MCR_d(c_i) \leq n_2C$, where $n_1 = 0$ and $0 < n_2 \leq 1$.
   This constraint says that the concepts that we choose should have rank less than the n_2 fraction of the maximum rank C (recall that C is the total number of concepts that we select, i.e., C is the maximum rank). This constraint selects concepts from the top of the ranked list.
2) **Middle:** $n_1C \leq MCR_d(c_i) \leq n_2C$, where $0 \leq n_1 \leq n_2 \leq 1$.
   This constraint selects concepts from the middle of the ranked list bounded by n_1C and n_2C.
3) **Lower:** $n_1C \leq MCR_d(c_i) \leq n_2C$, where $0 \leq n_1 < 1$ and $n_2 = 1$.
   This constraint selects concepts from the bottom of the ranked list bounded by n_1C and C.

In Section IV, we will show through human evaluation which part of the ranked list provides best replacements (in terms of generating believable fake documents).

The significance of other constraints in Equation 2 is as follows. The second constraint indicates that there should be at least one fake document generated from the original document. The third constraint indicates that the sum of the costs of substituting C_j concepts in d with their alternatives to generate d'_j from d should be bounded by $\mathcal{B}_j$, the budget assigned to generate d'_j. Note that for simplicity, we assign equal budget to each fake document (i.e., $\mathcal{B}_j = \frac{\mathcal{B}}{n}$). The fourth constraint implies that the total number of concepts substituted in d to generate different fake documents should not be more than C. The fifth constraint uses a boolean integer variable X_{kj}. $X_{kj} = 1$ indicates that concept c_k has been replaced to generate fake document d'_j. This constraint says that when k is fixed, the X_{kj}'s must add up to one. In other words, a given concept c_k can only be replaced to generate only one fake document, not many. However, if a single concept is allowed to be replaced multiple times to generate different fake documents, some of the other substituted concepts accompanying the given concept might also be selected again in order to satisfy the budget. This may produce (near-similar) fake documents with very minimal difference in the content. We intended to avoid this situation. However, one may set a bound on the number of times a concept is allowed to be replaced (instead of at most one as we do) such that $\sum_{j=1} X_{kj} \leq \mu$, where μ is the selected bound. This will not effect our formulation mentioned in Equation 2; however it may include this additional constraint involving μ. Note that a fake document can contain multiple substitutions depending upon the budget. If the budget is high, our system can allow multiple substitutions inside a single document.

We can now formally define the fake document generation problem as follows:

Definition III.8 (Fake Document Generation Problem)**.** *We are given an original document* d *and a budget* $\mathcal{B}$*. We extract a set of concepts* $c_1, c_2, \ldots$*. Each concept* C_i *has two attributes: its meta-centrality rank* $MCR(C_i)$ *and a cost of replacing it by an alternative concept* $Cost(c_i, ac_i)$*. The problem is to generate* n *fake documents from* d *by replacing a set of* C *concepts in such a way that the sum of the meta-centrality ranking of the selected concept will be minimum and the substitution cost will remain bounded within a certain budget* $\mathcal{B}_n$ *(where* $\mathcal{B}_n = \frac{\mathcal{B}}{n}$*) per fake document. Each concept can be* *replaced at most once to generate fake documents.*

The following result shows that computing an optimal set of fake documents is intractable.

Theorem 1. *The fake document generation problem (2) is NP-complete.*

Proof. We will prove this theorem by showing that (i) (2) is in the class NP, and (ii) there exists a polynomial time algorithm that reduces an instance of multiple knapsack problem (a known NP-complete problem) to an instance of fake document generation problem. Note that generating each fake document corresponds to a 0/1 knapsack problem.

- *0/1 knapsack problem*:
  Instance: Each item i has two attributes: a non-negative weight $W(i)$ and a non-negative value $V(i)$. There is a knapsack with weight t.
  Problem: Is there a subset of items with total weight at most T, such that the sum of their values is at least M?

Fake document generation problem is NP: Given a set with (say,) n' concepts, it is very easy to check if the sum of their replaceable costs is at most B_i and if the sum of corresponding semantic ranking is at most K. It takes linear time to add the costs and semantic rankings of all the items to find the *true/false* result. So the decision problem is linear in time. Therefore, the fake document generation problem is in NP.

0/1 Knapsack problem is polynomially reducible to the Fake document generation problem: We can reduce an instance of knapsack problem to an instance of fake document generation problem. We need to create such a knapsack problem that

$$\begin{cases} Cost(c_i) = W(i) \\ MCR(c_i) = V(i) \\ B = t \end{cases} \tag{3}$$

We have to show that the *Yes/No* answer of the new problem corresponds to the same answer to the original problem. The following deduction implies the new problem is equivalent to the original problem:

$$\begin{cases} \sum_{c_i \in C} Cost(c_i, ac_i) \leq B \Longleftrightarrow \sum_{i \in S} W(i) \leq T \\ \sum_{c_i \in C} MCR(c_i, ac_i) \leq K \Longleftrightarrow \sum_{i \in S} V(i) \geq M \end{cases} \tag{4}$$

Suppose we have a *Yes* answer to the new problem. This indicates that we can find a subset C of concepts that satisfy the left part of (4). Then this is also solution to the right part. So we must also have a *Yes* answer to the original problem. Conversely, if we have a *No* answer, it implies there is no subset C that satisfies the left part. So, the answer to the original problem must also be *No*. And this reduction can be done in polynomial time.

Therefore, the fake document generation problem (2) is NP-complete. □

Since the fake document generation problem is computationally intractable, we adopted the standard knapsack solver available in CPLEX[8] [27] to execute FORGE. For this, we map the fake document generation problem to an ordinary knapsack problem. Given a knapsack with a maximum weight T and a set of items C with different weights and costs ($W(i)$ and $V(i)$ indicate the weight and the value of item i respectively), the problem is to select a subset of items C' such that the sum of values of the selected items inside the knapsack is maximized (i.e., $maximize \sum_{i \in C' \subseteq C} V(i)$), and the sum of weights of the selected items is less than the weight of the knapsack (i.e., $\sum_{i \in C'} \subseteq W(i) \leq T$). Therefore, the mapping is as follows:

- A knapsack corresponds to a fake document.
- An item corresponds to a concept.
- We set the value $V(i)$ of an item i to $-MCR(c)$ (i.e., negative of $MCR(c)$) of a concept c. Note that we use $-MCR(V_i)$ because our objective function seeks to minimize the sum of the $MCR(c)$'s. Maximizing the sum of the $-MCR(c)$ values achieves the same optimal solution and is compatible with the requirement of maximization in knapsack problems.
- The weight of a knapsack corresponds to the budget assigned to generate the corresponding fake document.

Example III.6. *We now revisit the small sample paragraph shown in Example III.1. Here are two top fake versions of this paragraph that were generated.* We underline places where terms from the paragraph were changed.

> A $\underline{\text{surfactin}}$ composition for agricultural chemicals containing fatty acid polyoxyalkylene alkyl $\underline{\text{ester}}$ expressed by the following formula (I):
>
> $$R^1CO(EO)_{rn}(PO)_nOR^2 \ (I)$$
>
> wherein the fatty acid polyoxyalkylene alkyl ether has a narrow ratio of 55% by mass or more, where the narrow ratio is expressed by the following formula:
>
> $$Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i \ (A)$$

> A surfactant composition for agricultural chemicals containing fatty acid polyoxyalkylene alkyl $\underline{\text{ester}}$ expressed by the following formula (I):
>
> $$R^1CO(EO)_{rn}(PO)_n\underline{ROOR} \ (I)$$
>
> wherein the fatty acid polyoxyalkylene alkyl $\underline{\text{ester}}$ has a narrow ratio of 55% by mass or more, where the narrow ratio is expressed by the following formula:
>
> $$Narrow\ ratio = \sum_{i=n_{MAX-2}}^{i=n_{MAX+2}} Y_i \ (A)$$

E. Online Learning

The final module of FORGE is online learning where we incorporate feedback from security officers. A security officer can accept or reject a suggested concept(s) for replacement as well as their corresponding alternative concepts. To generate a fake document d' corresponding to an original document d, if the expert thinks that the candidate replaceable concepts $\{c_1, c_2, \cdots, c_k\}$ are all wrongly identified and should not be

[8] http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud

replaced, we insert the following additional constraint into the optimization problem (2):

$$X_{1j} + X_{2j} + \cdots + X_{kj} = 0$$

Similarly, given a set C of candidate replaceable concepts $\{c_1, c_2, \cdots, c_k\}$ for fake document d', if the expert suggests that any one among this set can be chosen because replacing all the concepts in this set with the corresponding alternative concepts might cause the resulting fake document to lack credibility, we insert the following new constraint:

$$X_{1j} + X_{2j} + \cdots + X_{kj} = 1$$

We then rerun FORGE to solve the fake document generation problem taking the user input into account.

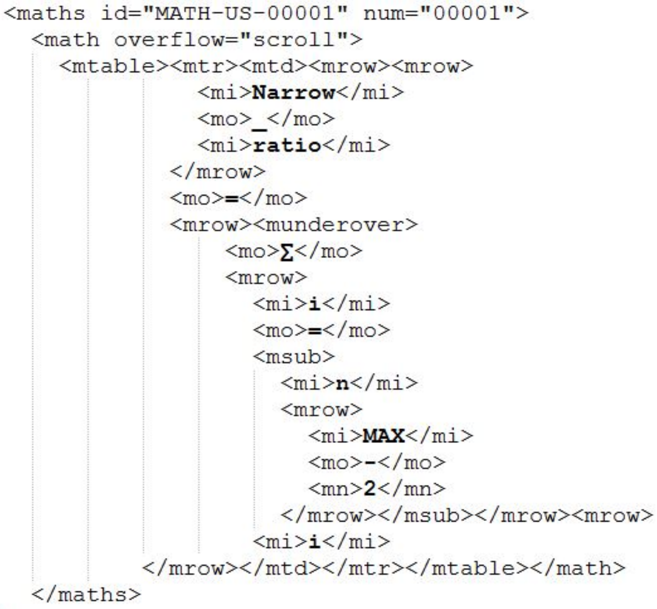
If the expert does not like the suggested alternative candidate concept ac_i for a certain concept c_i, s/he may suggest a completely new alternative concept ac'_i (that may not be present in the ontology). In this case, FORGE accepts this substitution as a valid substitution with minimum substitution cost, i.e., $Cost(c_i, ac'_i) = min_{j \neq i}\ Cost(c_j, ac_j)$, assuming that the expert's opinion is the best (least expensive) option for this substitution.

IV. Experimental Setup

In this section, we describe the dataset used in our experiments followed by the experimental setup.

A. Datasets Description

We created two datasets by crawling 250 documents each in the domains of agricultural chemistry (**AgChem** dataset) and computer science (**CSDocs** dataset). We then extracted some specific information (such as formula, mathematical equation, chemical agent etc.) from each document using the *HTML* tags present in the crawled documents (an example document is shown in Figure 5). We generated fake versions of each of these documents using a diverse set of parameter settings.



```
<maths id="MATH-US-00001" num="00001">
  <math overflow="scroll">
    <mtable><mtr><mtd><mrow><mrow>
              <mi>Narrow</mi>
              <mo>_</mo>
              <mi>ratio</mi>
          </mrow>
          <mo>=</mo>
          <mrow><munderover>
                <mo>∑</mo>
                <mrow>
                  <mi>i</mi>
                  <mo>=</mo>
                  <msub>
                    <mi>n</mi>
                    <mrow>
                      <mi>MAX</mi>
                      <mo>-</mo>
                      <mn>2</mn>
                    </mrow></msub></mrow><mrow>
                  <mi>i</mi>
        </mrow></mtd></mtr></mtable></math>
  </maths>
```

Fig. 5: An example document [15] with various HTML tags.

B. Parameter Settings

We now present the possible set of parameters of our system to conduct the experiments.

- **Meta-centrality:** We vary the following parameters in Equation 1:
  - $CM_s(v_i)$: the centrality of a vertex v_i in the document layer is calculated separately by the following four ways – degree centrality (DC), betweenness centrality (BC), closeness centrality (CC) and PageRank (PR) (as mentioned in Section III-C).
  - We vary β from 0.1 to 0.3 in increments of 0.05.
- **Ontological Distance:** We consider two types of ontological distance to measure $dist_{\mathcal{O}}(a, b)$ (where a and b are two concepts in the ontology):
  - *Graph-based*: Graph-based shortest path distance between a and b in the ontology.
  - *Entropy-based*: $max_{c \in S(a,b)}[-\log P(c)]$, where $S(a, b)$ is the set of all neighbors of both a and b, and $P(c)$ is the probability of occurrence of c in a specific corpus [28].
- **Budget:** We set $\mathcal{B} = 10$ as default. As we increase the budget, we will be able to generate more fake documents. We further assume that $\mathcal{B}$ is uniformly divided to generate n fake documents (i.e., $\mathcal{B}_j = \frac{\mathcal{B}}{n},\ \forall j = 1 : n$).
- **Other Parameters:** Two other parameters n_1 and n_2 used in the objective function (for dividing the ranked list into three buckets) are varied as follows: Upper ($n_1 = 0$, n_2 is varied from 0.1 to 0.3), Middle (n_1 is varied from 0.1 to 0.3 and n_2 is varied from 0.4 to 0.6), and Lower (n_1 is varied from 0.4 to 0.7, $n_2 = 1$). In each case, the increment is set to 0.1.

We choose default budget (i.e., $\mathcal{B} = 10$) and vary other parameters to generate fake documents with different parameter settings.

V. Experimental Results

A. Human Evaluation

Our human evaluation was done on two datasets. The **AgChem** dataset consists of 50 original documents, while the **CSDocs** dataset consists of 50 original docs. In both cases, for each document, we generated fake documents using all possible sets of parameters keeping the overall budget $\mathcal{B}$ default (one fake document was generated for each parameter setting). Since identifying a document as original or fake requires manual intervention, we requested 20 Engineering Masters students to evaluate FORGE for the **AgChem** dataset and 10 CS Masters students for the **CSDocs** dataset. Each human subject was shown a list of names corresponding to the documents, and was asked to select each name one by one. Upon selecting a name, the original document and its associated fake versions generated by different parameter settings of FORGE were shown to the human subject. The

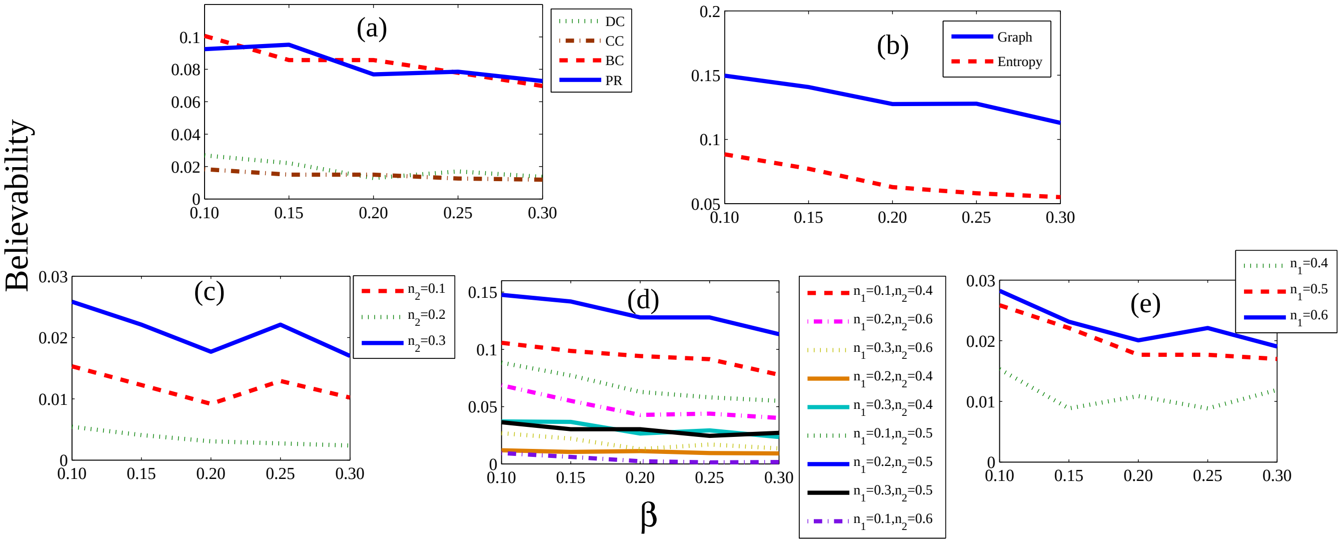


Fig. 6: Believability corresponding to different parameter settings of FORGE for AgChem dataset. In x-axis, we vary β and plot the believability for different types of (a) centrality measures, (b) distance measures, and different combinations of n_1 and n_2 for (c) upper, (b) middle and (c) lower buckets. The best parameter for FORGE is $\langle\beta = 0.1$, PageRank, graph-based distance, middle bucket with $n_1 = 0.2$ and $n_2 = 0.5\rangle$. The value corresponding to one parameter is reported by averaging the values for all possible combinations of other parameters.

subject was unaware about the originality of the documents. The task of the subject was to find the original document from the bunch. For a given document, each subject was asked to *select the top 3 documents* they felt were the original document, and rank them based on subject's confidence of the document being original. From this evaluation, we obtain the best parameter setting for FORGE and the overall performance.

B. Experimental Results

For each evaluation, we generate a matrix, where the rows correspond to the human subjects and the columns correspond to different parameter settings. Each entry (i, j) in this matrix corresponds to the fraction of times documents generated using parameter j is selected as the original document by subject i. We measure the overall performance of FORGE by the following metric:

Definition V.1 (Deception Factor)**.** *The deception factor DF of a fake document generation system is measured by the percentage of times a human subject has wrongly identified a (fake) document as the original document.*

Let $\{d_0^1, d_0^2, \cdots, d_0^m\}$ be m original documents for which we want to generate fake documents. For each original document d_0^i, let $\{d_1^{i'}, d_2^{i'}, \cdots, d_n^{i'}\}$ be n fake documents generated by FORGE. From the set $\{d_0^i, d_1^i, d_2^i, \cdots, d_n^i\}$, let us assume that a human subject identifies d_*^i as the original document. Then DF is calculated as:

$$DF = \frac{\sum_{i=1}^{m} 1 - \delta(d_*^i, d_0^i)}{m} \tag{5}$$

where $\delta(d_*^i, d_0^i) = 1$, if $d_*^i = d_0^i$, 0 otherwise. The greater the value of DF, the more efficient our system is at deception. In

For each human subject, we measure DF of our system at four levels: (i) *Level 0*: ignoring the rank of 3 choices per original document and declaring the user's guess to be correct if ANY of his three choices is correct, (ii) *Level 1*: considering only the first choice, (iii) *Level 2*: considering only the second choice, and (iv) *Level 3*: considering only the third choice. Tables IV and V report the DF (in %) of our system by averaging it in two different ways – (i) averaged over all the human subjects and its standard deviation, (ii) averaged over all documents and its standard deviation. The average in both cases is the same; however the standard deviation varies and remains significantly low in both cases (see Table IV). Human-level variation in terms of standard deviation is lower than that of document-level, indicating that human subjects more or less agreed with their annotations. *Note that the deception factor for a user can vary from 0 to 1. For example, at level 0, the user is selecting 3 documents in the hope that one of them is the correct one. If one of them is correct, he is considered to have returned the correct answer even though the other two are wrong.* The experiment is not one where the user is presented three choices and asked to pick the right one in which case there would be a prior probability of 2/3 that he gets it wrong.

TABLE IV: **AgChem** Dataset. Deception factor (average and standard deviation (SD)) of FORGE in four levels and two different averaging strategies.

(a) Avg. over human subjects

DF	Average	SD
Level 0	94.3%	0.029
Level 1	97.4%	0.018
Level 2	98.3%	0.019
Level 3	98.6%	0.013

(b) Avg. over all documents

DF	Average	SD
Level 0	94.3%	0.053
Level 1	97.4%	0.045
Level 2	98.3%	0.034
Level 3	98.6%	0.034

TABLE V: **CSDocs** Dataset. Deception factor (average and standard deviation (SD)) of FORGE in four levels and two different averaging strategies.

(a) Avg. over human subjects		
DF	Average	SD
Level 0	92.2%	0.037
Level 1	95.8%	0.030
Level 2	97.8%	0.015
Level 3	98.6%	0.020

(b) Avg. over all documents		
DF	Average	SD
Level 0	92.2%	0.051
Level 1	95.8%	0.047
Level 2	97.8%	0.039
Level 3	98.6%	0.035

In order to obtain the best parameter setting for FORGE, we define the following metric:

Definition V.2 (Believability)**.** *The believability of a parameter setting is the fraction of times fake documents generated by this setting are chosen as original by human subjects.*

Let us consider the same set of notations used in Equation 5. Further assume that for each original document d_0^i, d_j^i is the fake document generated by j^{th} parameter setting. The believability $\mathcal{B}$ of the j^{th} parameter setting is defined as:

$$\mathcal{B}(j) = \frac{\sum_{h \in \mathcal{H}} \sum_{i=1}^{m} \delta(d_{h*}^i, d_j^i)}{m|\mathcal{H}|} \tag{6}$$

where $\mathcal{H}$ is the set of human subjects, and d_{h*}^i is the document identified by human subject h as the original document corresponding to the i^{th} original document d_0^i, and $\delta(d_{h*}^i, d_j^i) = 1$ if h chooses d_j^i as the original document. The higher the believability of a parameter setting, the better the parameter setting to generate the believable fake documents.

We again ignore the rank of the 3 top choices per original document and consider them equally. We then vary each type of parameter and report its believability by averaging the values for all possible combinations of other parameters. Figure IV shows the results on the **AgChem** dataset. Figure IV(a) uses different centrality measures and shows that as β increases, PageRank and betweenness centrality outperform others by a significant margin – out of five different values of β, PageRank beats betweenness centrality in three cases. Figure IV(b) uses two different ontological distance metrics and shows that graph-based distance yields better performance. Furthermore, we consider different combinations of n_1 and n_2 for the three buckets (Upper, Middle, Lower) in the fake document generation problem. For first, second and third buckets, best result is obtained with $n_1 = 0$ and $n_2 = 0.3$ (Figure IV(c)); $n_1 = 0.3$ and $n_2 = 0.4$ (Figure IV(d)); and $n_1 = 0.8$ and $n_2 = 1$ (Figure IV(e)) respectively. However, amongst all of them, FORGE performs the best with $n_1 = 0.2$ and $n_2 = 0.5$. Therefore, in the rest of the experiments, we choose the following parameter setting for FORGE as default: $\langle\beta = 0.1$, PageRank, graph-based distance, middle bucket with $n_1 = 0.2$ and $n_2 = 0.5\rangle$.

Figure 7 shows the same on the **CSDocs** dataset Different centrality measures are compared in Figure 7a, showing that PageRank and Betweenness centrality outperform others by a significant margin on this dataset too. Figure 7b shows again that that Graph-based distance yields better performance than entropy-based distance. Furthermore, Figures 7c, 7d and 7e describe show performance using different combination of n_1 and n_2 for the three buckets considered. In particular for the first, second and third bucket, the best results are obtained respectively with $n_1 = 0$ and $n_2 = 0.2$ (Figure 7c); $n_1 = 0.2$ and $n_2 = 0.5$ (Figure 7e) and $n_1 = 0.6$ and $n_2 = 1$ (Figure 7d).

A critic may argue that the deception factor of our system is high because of the large number of fake documents generated which in turn decreases the a priori probability of picking the original document from the lot. While it is true that increasing the number of fake documents indeed makes it harder for the attacker to correctly identify the original document, we argue that it is also the quality of the fake document which makes the attacker believe that the fake document is original.

We therefore hypothesized that a system which generates fake documents by randomly replacing concepts with other concepts has a lower deception factor than FORGE. To validate our hypothesis, we designed a random fake document generation system (called R_FORGE) which, given an original document, randomly chooses a certain number of concepts (as mentioned in Section III-A). For each chosen concept, R_FORGE leverages WordNet[9] to generate an alternative concept to replace it. For instance, let "surface culture process" be a replaceable concept for which we want to generate an alternative concept using WordNet. Since the entire phrase is not present in WordNet, we first break it into possible small pieces such as "surface culture", "culture process", "process" etc. One of these small phrases might be directly found in WordNet. We then replace the portion with its synonym. In our case, "operation" is a synonym of "process", and we generate "surface culture operation" as an alternative concept.

For each original document, we generated 3 fake document using FORGE with the best parameter settings and 3 fake documents using R_FORGE. We also made sure that the same number of concepts have been replaced by both the systems. Therefore, 6 fake documents were generated for each of 50 original documents. The 20 human subjects who participated in the previous **AgChem** evaluation were further requested to evaluate these two systems. Each human subject was given 6 fake documents and asked to select the original document (note that human subjects were unaware about the fact that the original document was not present in the lot). We intended to see if the fake documents generated by FORGE are of better quality than that of R_FORGE. Then for each subject, we measure the *credibility* ($\mathcal{C}$) of FORGE as follows.

Definition V.3 (Credibility)**.** *The credibility of* FORGE *is measured by the percentage of times a fake document generated* FORGE *is selected as original document.*

Let $\{d_0^1, d_0^2, \cdots, d_0^m\}$ be m original documents. For each original document d_0^i, let $\{d_0^{i1}, d_0^{i2}, d_0^{i3}\}$ be the fake documents generated by R_FORGE and $\{d_0^{i4}, d_0^{i5}, d_0^{i6}\}$ be the fake documents generated by FORGE. Then the credibility $\mathcal{CR}$ of FORGE is measured as:

$$\mathcal{CR} = \frac{\sum_{h \in \mathcal{H}} \sum_{i=1}^{m} \delta(d_{h*}^i, \{d_0^{i4}, d_0^{i5}, d_0^{i6}\})}{m|\mathcal{H}|} \tag{7}$$

[9] https://wordnet.princeton.edu/

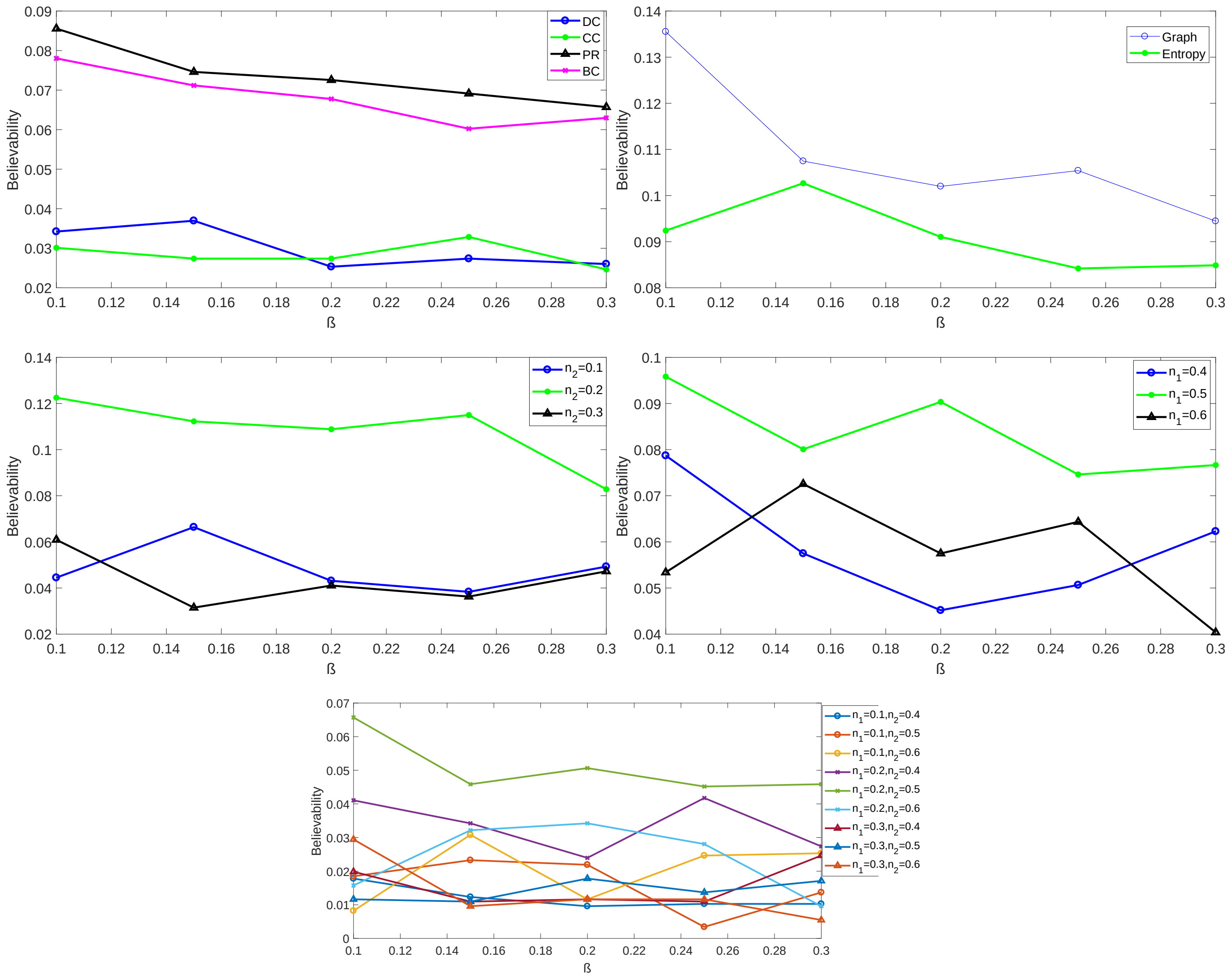


Fig. 7: Believability corresponding to different parameter settings of FORGE for CSDocs dataset. In x-axis, we vary β and plot the believability for different types of (a) centrality measures, (b) distance measures, and different combinations of n_1 and n_2 for (c) upper, (b) middle and (c) lower buckets. The best parameter for FORGE is $\langle\beta = 0.1$, PageRank, graph-based distance, middle bucket with $n_1 = 0.2$ and $n_2 = 0.4\rangle$. The value corresponding to one parameter is reported by averaging the values for all possible combinations of other parameters.

where $\mathcal{H}$ is the set of human subjects, and d^i_{h*} is the document chosen by human subject h as the original document corresponding to the i^{th} original document d^i_0. $\delta(d^i_{h*}, \{d^{i4}_0, d^{i5}_0, d^{i6}_0\}) = 1$ if $d^i_{h*} \in \{d^{i4}_0, d^{i5}_0, d^{i6}_0\}$, 0 otherwise. The higher the credibility, the better the quality of fake documents generated by FORGE compared to the randomly generated fake documents. Note that the formulation of two metrics – believability and credibility are same; however the former is specific w.r.t. a parameter setting, whereas the latter is specific to a system.

We observe that the credibility (in %) of FORGE is 98.20% (and standard deviation of 0.017). This result confirms our hypothesis that the high deception factor achieved by FORGE is not only due to the number of fake documents generated by FORGE but also the quality of the generated documents which appear believable to the attackers.

VI. Enabling Legitimate Users to Identify Original Documents

One problem with FORGE is that when a given repository $\mathcal{R}$ of documents is expanded via the addition of fake versions of those documents, legitimate users may not be able to identify the original documents. To address this, we can apply a cryptographic approach based on message authentication codes [29].

Roughly, a message authentication code consists of a pair of algorithms $(\mathsf{Mac}, \mathsf{Vrfy})$ where the authentication algorithm Mac takes as input a key k and a document d and outputs a tag t; we denote this by $t := \mathsf{Mac}_k(d)$. The verification algorithm Vrfy takes as input a key k, a document d, and a tag t and outputs a bit b, with $b = 1$ denoting accept and $b = 0$ denoting reject; we denote this by $b := \mathsf{Vrfy}_k(d, t)$. The basic correctness requirement is that for any key k and document d, if $\mathsf{Mac}_k(d)$ outputs t, then $\mathsf{Vrfy}_k(d,t) = 1$.

For our purposes, we require a message authentication code satisfying the following security property: if k is random and unknown to the attacker, then for any set of documents $d_1, d_2, \ldots$, the tags $t_1 := \mathsf{Mac}_k(d_1), t_2 := \mathsf{Mac}_k(d_2), \ldots$ are indistinguishable from a sequence of uniform and independent values. (That is, Mac is a variable-input-length pseudorandom function.) Although this is stronger than the standard security requirement for message authentication codes, many widely used and standardized message authentication codes (e.g., CBC-MAC or HMAC) satisfy this security definition under standard assumptions.

Given $(\mathsf{Mac}, \mathsf{Vrfy})$, we now show how to address our original problem. First we generate a random, long-term key k that will be given to every legitimate user. We then modify the process of generating fake documents, as follows. Given an original document d, we first compute $Fake(d) = \{d'_1, \ldots, d'_n\}$ as before. We then associate a "marker" with each document: the marker of an original document d is a correctly computed tag $\mathsf{Mac}_k(d)$, but every fake document is marked with a random string of the appropriate length.

Given a repository $\mathcal{R}^*$ containing both original and fake documents along with their associated markers, a legitimate user (who knows the key k) can identify the original documents by looking for valid tags: given a document/marker pair (d,t), the user assumes that d is an original document if and only if $\mathsf{Vrfy}_k(d,t) = 1$.

We now argue that this scheme is both *correct* and *secure*:

Correctness. We need to show that the original documents are correctly identified as such by legitimate users, and that fake documents are not incorrectly assumed to be original. The former holds by correctness of the message authentication code. The latter property holds because the probability that a random string is equal to a valid tag is negligible. (E.g., HMAC-SHA1 has a 160-bit tag, and so the probability that a random string is equal to the correct tag for some fake document d' is 2^{-160}.)

Security. We also need to argue that adding "markers" to the documents does not make it any easier for an attacker to distinguish fake documents from original documents. This holds due to the security of the message authentication code, namely, the property that a legitimate tag on a document d is indistinguishable (for an attacker who does not know k) from a random string.

VII. Future Work

In general, a document such as a patent consists of variety of different types of information such as text, equations/formulas, tables, images, and diagrams. In this paper, we focus primarily on modifying the text though some equations and chemical formulas are also modified. However, a diagram (e.g. a flowchart or a block diagram) can contain information about the text - the same is true of figures, tables etc. In order to *consistently* change different types of entities within a document, a single representational framework is needed to express the content in these different parts of a document, before we can understand how changes in one part of a document affect another. For instance, consider an equation in the original document of the form $y = 3x^2 + 5$. If a fake version of the document were to change the square to a cube, resulting in the (fake) equation $y = 3x^3 + 5$, this would be believable as long as the original document did not, for instance, discuss the equation in the text and say something along the lines of ".... y is quadratic in x...". In this case, to maintain consistency, the fake version of the document would need to change the word "quadratic" to "cubic" — something that is obviously impossible to do without some notion of semantic consistency. This is an important issue to address in future work.

A second potential avenue for future work is the use of an ontology to identify a possible replacement for a concept. For instance, many good ontologies exist — but there is a question of whether these existing ontologies are good enough for the purpose of generating fakes. Generating new ontologies which are good enough can also pose a challenge. Thus, a possible replacement for ontologies within FORGE or other similar frameworks would be an important avenue for exploration.

VIII. Conclusion

In today's world, large enterprises can assume that they will be hacked by adveraries who are determined to steal a variety of information that could include personally identifying information, financial/bank information, and intellectual property, amongst others. The majority of work in cybersecurity has focused on keeping intruders and malicious actors out of networks, e.g. by developing methods to detect malicious connections or traffic. Much less work has gone into handling attacks that penetrate a network successfully - in other words, how does one defend an enterprise's IP assets when one does not even know that it has been compromised?

This paper suggests an initial approach to this problem in the case when the IP assets of the organization are of a technical nature (e.g. patents, engineering designs). Our research has three major innovations. First, we show that a technical document archive can be automatically represented as a multi-layer graph via simple, off the shelf natural language processing techniques — the use of MLGs to represent document content for security purposes is new. Second, we propose the important concept of Meta-Centrality for MLGs that extends existing centrality measures for ordinary graphs to a family of measures of MLGs. Third, we show that the problem of generating fake documents is an optimization problem and further show that it is NP-hard: we then show that we can solve it in practice using approximate algorithms to solve the knapsack problem. Fourth, we show experimentally using two datasets, one involving a panel of 20 human subjects and another involving a panel of 10 human subjects, that our methods are capable of achieving high levels of deception. Finally, we address the problem of a legitimate user who needs to access the legitimate (non-fake) documents in the system and propose a simple cryptographic approach based on method authentication codes.

Of course, there is much further work to be done. Our paper focuses on technical documents — but similar methods can be studied for other types of documents such as those involving financial information, business plans, and personally identifying information, amongst others.

ACKNOWLEDGMENT

Parts of this work were funded by ONR Grants N00014-16-1-2896 and N00014-15-1-2007, N00014-18-1-2670 and ARO Grants W911NF-13-1-0421 and W911NF-15-1-0576 and W911NF-14-1-0358, and Maryland Procurement Office grant H9823014C0137. Tanmoy would like to acknowledge the support of two SERB, DST Grants – ECR/2017/001691, Ramanujan Faculty Award along with the support of the Infosys center of AI at IIIT Delhi.

REFERENCES

[1] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering on multi-layer graphs via subspace analysis on grassmann manifolds," *IEEE Transactions on signal processing*, vol. 62, no. 4, pp. 905–918, 2014.

[2] J. Yuill, M. Zappe, D. Denning, and F. Feer, "Honeyfiles: deceptive files for intrusion detection," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.*, June 2004, pp. 116–122.

[3] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2009, pp. 51–70.

[4] Y. Park and S. J. Stolfo, "Software decoys for insider threat," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. ACM, 2012, pp. 93–94.

[5] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012, pp. 461–475.

[6] B. Whitham, "Automating the generation of fake documents to detect network intruders," *nt. J. Cyber-Security Digit l. Forensics*, vol. 10, no. 2, pp. 103–118, 2013.

[7] J. White and D. R. Thompson, "Using synthetic decoys to digitally watermark personally-identifying data and to promote data security," in *in Proceedings of the International Conference on Security and Management (SAM)*, 2006, pp. 91–99.

[8] L. Wang, C. Li, Q. Tan, and X. Wang, *Generation and Distribution of Decoy Document System*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 123–129.

[9] M. Bercovitch, M. Renford, L. Hasson, A. Shabtai, L. Rokach, and Y. Elovici, "Honeygen: An automated honeytokens generator." in *ISI*. IEEE, 2011, pp. 131–136. [Online]. Available: http://dblp.uni-trier.de/db/conf/isi/isi2011.html#BercovitchRHSRE11

[10] A. Shabtai, Y. Elovici, and L. Rokach, *A Survey of Data Leakage Detection and Prevention Solutions*. Springer Publishing Company, Incorporated, 2012.

[11] D. Kushner, "Digital decoys [fake mp3 song files to deter music pirating]," *IEEE Spectrum*, vol. 40, no. 5, p. 27, May 2003.

[12] J. Voris, N. Boggs, and S. J. Stolfo, "Lost in translation: Improving decoy documents via automated translation," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*. IEEE, 2012, pp. 129–133.

[13] B. Whitham, "Towards a set of metrics to guide the generation of fake computer file systems," in *Proceedings in the 15th Australian Information Warfare Conference*, Perth, Australia, 2014.

[14] ——, "Design requirements for generating deceptive content to protect document repositories," in *Proceedings in the 15th Australian Information Warfare Conference*, Perth, Australia, 2014.

[15] A. Kanetani, H. Izumoto, S. Sato, and T. Kano, "Surfactant composition for agricultural chemicals," May 23 2013, uS Patent App. 13/741,708. [Online]. Available: https://www.google.com/patents/US20130131363

[16] C. D. Paice and P. A. Jones, "The identification of important concepts in highly structured technical papers," in *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1993, pp. 69–78.

[17] P. Vossen, "Extending, trimming and fusing wordnet for technical documents." The Association for Computational Linguistics, 2001.

[18] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, vol. 2, no. 3, p. 203, 2014.

[19] J. R. Firth, "A synopsis of linguistic theory 1930-55." vol. 1952-59, pp. 1–32, 1957.

[20] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[21] Y.-Y. Chen, Q. Gan, and T. Suel, "Local methods for estimating pagerank values," in *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, ser. CIKM '04. New York, NY, USA: ACM, 2004, pp. 381–389. [Online]. Available: http://doi.acm.org/10.1145/1031171.1031248

[22] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010.

[23] J. Jung and J. Euzenat, "Measuring semantic centrality based on building consensual ontology on social network," in *Proc. 2nd ESWS workshop on semantic network analysis (SNA)*. No commercial editor., 2006, pp. 27–39.

[24] D. Leprovost, L. Abrouk, N. Cullot, and D. Gross-Amblard, "Temporal semantic centrality for the analysis of communication networks," in *International Conference on Web Engineering*. Springer, 2012, pp. 177–184.

[25] M. C. Traub, M. H. Lamers, and W. Walter, "A semantic centrality measure for finding the most trustworthy account," in *Proceedings of the IADIS international conference informatics*, 2010, pp. 117–125.

[26] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, Jun. 1993. [Online]. Available: http://dx.doi.org/10.1006/knac.1993.1008

[27] "IBM ILOG CPLEX Optimizer," urlhttp://www-01.ibm.com/software/integration/optimization/cplex-optimizer/, Last 2010.

[28] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research*, vol. 11, pp. 95–130, 1998. [Online]. Available: http://www.cs.washington.edu/research/jair/abstracts/resnik99a.html

[29] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd ed. Chapman & Hall/CRC Press, 2014.

Tanmoy Chakraborty is an Assistant Professor and a Ramanujan Fellow at the Dept. of Computer Science & Engg., IIIT Delhi, India. Prior to this, he was a postdoctoral researcher at University of Maryland, College Park, USA. He completed his Ph.D as a Google India PhD fellow at IIT Kharagpur, India in 2015. His primary research interests include social network analysis, Data Mining, and Natural Language Processing. He has authored more than 90 publications in refereed journals, book chapters, and conferences. He has received several awards including the Google India Faculty Award, Early Career Research Award (DST, India), DAAD Faculty award, Best reviewer award in WWW'18, best PhD thesis award by Xerox Research, IBM Research and Indian National Academy of Engineering (INAE). http://faculty.iiitd.ac.in/~tanmoy/

Sushil Jajodia is University Professor, BDM International Professor and Director of Center for Secure Information Systems at George Mason University. Prior to joining Mason, he held permanent positions at NSF, NRL, and University of Missouri-Columbia. He has sustained a highly active research agenda spanning database and cyber security for over 30 years. According to the Google Scholar, he has over 42,000 citations and his h-index is 103.

Jonathan Katz is a professor in the Department of Computer Science at the University of Maryland, where he also serves as director of the Maryland Cybersecurity Center. His research interests are in cryptography, privacy, and the science of cybersecurity.

Antonio Picariello is a Full Professor in the Department of Electrical Engineering and Information Technology, University of Naples Federico II. He got a Ph. D. in Computer Engineering at the University of Napoli Federico II. He is the director of the National Lab of Computer Science, Telematics and Multimedia (ITEM) of the Italian Consortium on Computer Science and Engineering (CINI). He works in the field of Multimedia Database and Multimedia Information Systems, Multimedia Ontology and Semantic Web, Natural Language Processing, Big Data, Big Data analytics and Social Networks Analysis.

Giancarlo Sperli is a Researcher at the Consorzio Interuniversitario per l'Informatica (CINI). He hold a Master's Degree and a Bachelor's Degree in Computer Science and Engineering, both from the University of Naples Federico II and in 2018 he received the Ph.D degree in Information Technology and Electrical Engineering of University of Naples "Federico II". His main research interests are in the area of Cybersecurity, Semantic Analysis of Multimedia Data and Social Networks Analysis.

V.S. Subrahmanian is the Dartmouth College Distinguished Professor in Cybersecurity, Technology, and Society and Director of the Institute for Security, Technology, and Society at Dartmouth. He previously served as a Professor of Computer Science at the University of Maryland from 1989-2017 where he created and headed both the Lab for Computational Cultural Dynamics and the Center for Digital International Governmen. He also served for 6+ years as Director of the University of Maryland's Institute for Advanced Computer Studies. An elected fellow of both AAAI and AAAS, he has developed methods to analyze text/geospatial/relational/social network data, learn behavioral models from the data, forecast actions, and influence behaviors with applications to cybersecurity and counter-terrorism. He serves on the editorial boards of numerous journals including Science, the Board of Directors of the Development Gateway Foundation (set up by the World Bank), SentiMetrix, Inc., and on the Research Advisory Board of Tata Consultancy Services. He previously served on DARPA's Executive Advisory Council on Advanced Logistics and as an ad-hoc member of the US Air Force Science Advisory Board. Home page: https://www.cs.umd.edu/users/vs/