

# MetaPaper: Lessons from 1K rejected research papers

*Christos Faloutsos*

CMU

<https://www.cs.cmu.edu/~christos/MetaPaper>



# Disclaimer – meta-advice

Mainly, for ML / Data Science / DB venues

For other venues:

Some of the suggestions may still hold, but:

👉 Team up with seasoned authors from **those** venues



Lessons Learned



© 2021-2024 Christos Faloutsos



COMPUTER VISION  
GENERAL

# Self-introduction

## Qualifications:

<https://www.cs.cmu.edu/~christos/>



- Over 30+ years of research-paper submissions from CMU (and UofT, UMD, AT&T, IBM, MSR, etc)
- 3-digit acceptances
- 4-digit (~1K) rejections
- ...

# Self-introduction

## Qualifications:

<https://www.cs.cmu.edu/~christos/>



- Over 30+ years of research-paper submissions from CMU (and UofT, UMD, AT&T, IBM, MSR, etc)
- 3-digit acceptances
- 4-digit (~1K) rejections
- ... and counting (sigh!)

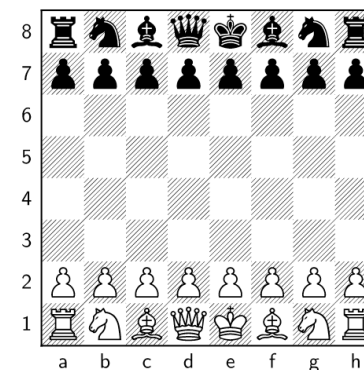
## On the bright side:

*You may learn much more from a game you lose than from a game you win.*

**José Raúl Capablanca**  
World chess champion 1921-27  
(‘the Cuban chess machine’)



[wikipedia.org](https://en.wikipedia.org/wiki/Jos%C3%A9_Ra%C3%9Ful_Capablanca)



## On the bright side:

*You may learn much more from a game you lose than from a game you win.*

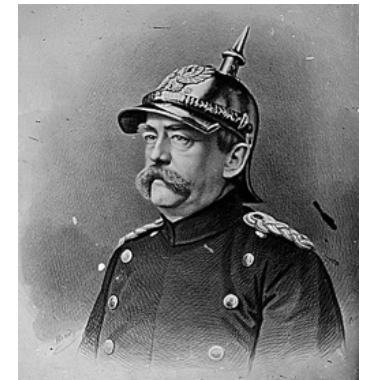


*Only a fool learns from his own mistakes. The wise man learns from the mistakes of others.*

(attr. to Bismarck ('Iron Chancellor') - [wikiquote.org](http://wikiquote.org))



[wikipedia.org](http://wikipedia.org)



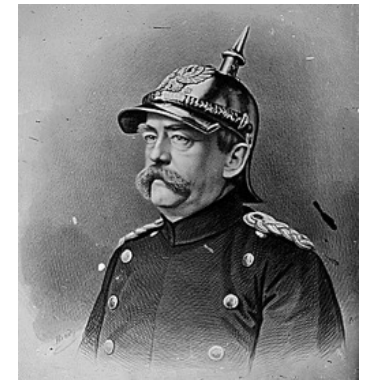
[germanculture.com.ua](http://germanculture.com.ua)

(fool?)



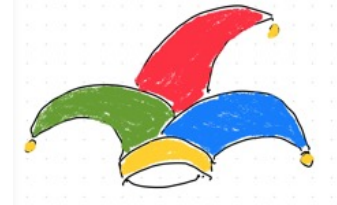
*Only a fool learns from his own mistakes. The wise man learns from the mistakes of others.*

(attr. to Bismarck ('Iron Chancellor') - [wikiquote.org](https://www.wikiquote.org/))



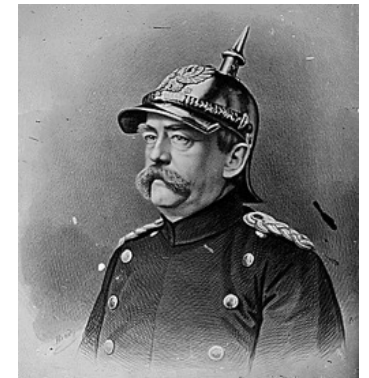
[germanculture.com.ua](https://germanculture.com.ua)

(ouch!)



*Only a fool learns from his own mistakes. The wise man learns from the mistakes of others.*

(attr. to Bismarck ('Iron Chancellor') - [wikiquote.org](https://www.wikiquote.org))



[germanculture.com.ua](https://germanculture.com.ua)

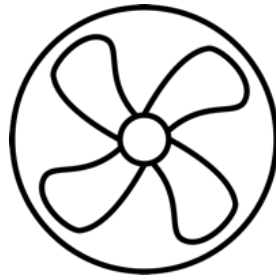


# Outline

- Top 3 lessons: ‘F.A.N.’



- F
- A
- N



- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions

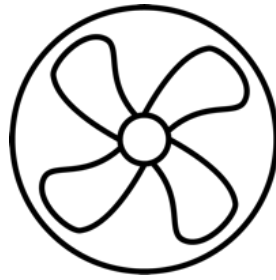


# Outline

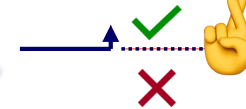
- Top 3 lessons: ‘F.A.N.’



– F



– A



- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions



# ‘F’ – which one?



- Pick one to review
- Q1) Which one?
- Q2) Why?

(a)

(b)

(c)

**C. Document Embedding and Clustering**

Much work has been done to represent documents in a machine-understandable format. The most widely-used approaches to represent documents include bag of words [1], and term frequency-inverse document frequency (tf-idf) [18]. These methods are commonly used for plagiarism detection [19], [20], [21], [22], which is a similar setting to near-duplicate detection. However, none of these methods do visualization or ranking, and some assumptions do not work in our case, i.e. [22] assumes documents consist of multiple lines, which is not the case for tweets or the majority of exact advertisements.

grammatical errors are common. Thus, these methods are not generalizable.

**E. Minimum Description Length**  
The Minimum Description Length principle (MDL) [37] assumes that the best model  $M \in \mathcal{M}$  for data  $D$  minimizes  $C(M) + C(D|M)$ , where  $C(\cdot)$  is defined as the cost, i.e. number of bits, needed to describe  $\cdot$  losslessly. The main insight is that it penalizes both the model cost  $C(M)$ , as well as the encoding of encodings from the model  $C(D|M)$  - while several other methods ignore the model complexity. MDL has been extremely successful in several data mining

**Abstract**

Given a million exact advertisements, how can we spot near-duplicates? Such micro-clusters of ads are usually signals of human trafficking. How can we summarize them, visualize, compare for enforcement to act? Can we build a general tool that works for different languages? Finding micro-clusters of near-duplicate documents is useful in multiple, additional settings, including spam-bot detection in Twitter ads, plagiarism, and more.

We present INFOSHIELD, which makes the following contributions: (a) *Practical*, being scalable and efficient on real data; (b) *Parameter-free and Principled*, requiring no user-defined parameters; (c) *Interpretable*, finding a document to be the cluster representative, highlighting of the common phrases, and automatically detecting “what”, i.e. phrases that differ in every document; and (d) *Generalizable*, being on multiple domains, specific methods in Twitter bot detection and human trafficking document comparison, as well as being language-independent, finding clusters in Spanish, Italian, and Japanese. Interpretability is particularly important for the anti-human-trafficking domain, where law enforcement must identify repeat ads.

Our experiments on real data show that INFOSHIELD correctly identifies Twitter bots with an F1 score over 90% and detects human-trafficking ads with 84% precision. Moreover, it is scalable, requiring about 1 hour for 2 million documents on a stock laptop.

**Index Terms**—Automated Detection, Text Mining, Clustering Methods, Minimum Description Length (MDL), Anti-Human Trafficking

**I. INTRODUCTION**

Given many documents, the majority of which do not belong to any cluster, how can we find small clusters of related documents? The driving application is human trafficking detection, where exact ads that are very similar are usually a sign of trafficking.

Finding related documents is a problem with numerous applications, such as search engines, plagiarism detection, mailing address de-duplication, and more.

In this paper, we develop INFOSHIELD, a general, information theory based method, and we illustrate its generality, effectiveness and scalability on two settings, tweets. Twitter: This particular application benefits from a vast amount of publicly available data.

**B. Application to Twitter Bot Detection**

Detection of organized activity has a clear application to bot detection; given millions of tweets, most of which come from legitimate users, how can we find tweets that exhibit bot-like behavior? The simplest kind of bot behavior is spamming, i.e. posting related documents in a problem with numerous applications, such as search engines, plagiarism detection, mailing address de-duplication, and more.

In this paper, we develop INFOSHIELD, a general, information theory based method, and we illustrate its generality, effectiveness and scalability on two settings, tweets. Twitter: This particular application benefits from a vast amount of publicly available data.

Dataset	Twitter Data				Human Trafficking Data				
	Docset	Time	Size	Pos. %	Docset	Time	Size	Pos. %	
Human	240	1hr	11.5MB	100%	Human	240	1hr	11.5MB	100%
Random	240	1hr	11.5MB	0%	Random	240	1hr	11.5MB	0%
Control (C)	240	1hr	11.5MB	0%	Control (C)	240	1hr	11.5MB	0%
Twitter (T)	240	1hr	11.5MB	0%	Twitter (T)	240	1hr	11.5MB	0%
Total (S)	240	1hr	11.5MB	0%	Total (S)	240	1hr	11.5MB	0%

TABLE VIII. INFOSHIELD performs well. Notice that INFOSHIELD beat or approaches the best domain-specific method in both settings. Bold shows the best score, underline shows methods within 10 points of the best. Methods in red are supervised, while INFOSHIELD is unsupervised.

Fig. 1. INFOSHIELD works: being precise (left), scalable (middle), and interpretable (right), detecting and visualizing clusters (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

**C. Our Method**

Our first insight is to formalize the problem with information theory, and use the Minimum Description Length (MDL) principle to find good templates, which represent cluster text, with “what”, i.e. parts of the template that differ for each document. We mark slots with red highlights in Figure 1 (right). We then use this summary to visualize the cluster.

INFOSHIELD is *parameter-free*, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the INFOSHIELD-FINE part of our method.

The second insight is a novel pre-processing method, INFOSHIELD-COARSE, that dramatically improves scalability to be quasi-linear, by (a) eliminating single-copy documents and (b) grouping the rest in coarse, but mainly homogeneous, clusters.

The resulting INFOSHIELD has a long list of desirable properties: It is

- *Practical*, processing 1 million documents within 2 hours on a Dell Alienware laptop with 32GB RAM and i7 processor running Arch Linux; and correctly identifying Twitter spambots with an F1 score of 90% or higher.

- *Parameter-free & Principled*, requiring no user-defined parameters thanks to the Minimum Description Language principle.
- *Interpretable*, providing a clear visualization and summarization of the discovered micro-clusters.
- *Generalizable and domain independent* – we show results on two diverse areas, namely, Twitter data, and HTdata; as well as on multiple languages, i.e. Spanish, Italian, English.

**Reproducibility:** Our code is open-sourced at <https://github.com/mengshih/InfoShield>. The HT dataset is available to researchers after NDA (email Dr. Cara Jones [caraj@manassasvtc.com](mailto:caraj@manassasvtc.com)). The Twitter datasets are publicly available (see [4]).

**II. BACKGROUND AND RELATED WORK**

There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

**A. Human Trafficking Detection**

Some previous works try to classify whether or not a particular advertisement is suspected of HT [5], [6], [7], [8]. For instance, HTDN [7] proposes a supervised deep multiclass model trained on 10K manually labeled ads. Their results are later improved, on the same data, using an off-the-shelf neural network [9]. Unfortunately, due to the adversarial nature of exact advertisements, these predefined or learned features don’t stay relevant over time. These labeled ads are also expensive to obtain (requiring the precious time of domain experts) and are error-prone, as will be discussed in Section V. Moreover, inspecting ads individually, we might overlook ads that are part of an organized activity but do not stand out on their own. Therefore, unsupervised algorithms that find groups of organized activity are preferred in this domain. Template Matching [10] exploits the above insight, being the first anti-HT method to use knowledge to perform clustering. However, the interpretability of clusters is limited, and the algorithm isn’t scalable.

**B. Social Media Bot Detection**

Most efforts in detecting bots in social media platforms are formulated as supervised classification based on features from users and the content they post [11], [12]. Fewer works look for anomalies or fraud in networks, rather than in text, for instance [13]. A notable method, Botometer [14], formerly called BotNois, is an online service that provides a score of likelihood that a particular user is a bot. Since it is the only state-of-the-art method with public access to the implementation, we will use it as a baseline for our experiments in Section V. [4] gives a more comprehensive overview of Twitter bot detection methods, and also provides the dataset we will use in Section V-A1.

Very few works focus on detecting organized activity: groups working together to mislead people about who they are and what they are doing, which is a rising issue [15]. ND-Syn [16] finds a related but different type of behavior, i.e. “retweet spam”, where groups of multiple users exhibit organized behavior by consistently retweeting a particular user’s tweets.

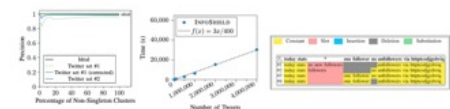


Fig. 1. INFOSHIELD works: being precise (left), scalable (middle), and interpretable (right), detecting and visualizing clusters (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

**C. Our Method**

Our first insight is to formalize the problem with information theory, and use the Minimum Description Length (MDL) principle to find good templates, which represent cluster text, with “what”, i.e. parts of the template that differ for each document. We mark slots with red highlights in Figure 1 (right). We then use this summary to visualize the cluster.

INFOSHIELD is *parameter-free*, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the INFOSHIELD-FINE part of our method.

The second insight is a novel pre-processing method, INFOSHIELD-COARSE, that dramatically improves scalability to be quasi-linear, by (a) eliminating single-copy documents and (b) grouping the rest in coarse, but mainly homogeneous, clusters.

The resulting INFOSHIELD has a long list of desirable properties: It is

- *Practical*, processing 1 million documents within 2 hours on a Dell Alienware laptop with 32GB RAM and i7 processor running Arch Linux; and correctly identifying Twitter spambots with an F1 score of 90% or higher.

- *Parameter-free & Principled*, requiring no user-defined parameters thanks to the Minimum Description Language principle.
- *Interpretable*, providing a clear visualization and summarization of the discovered micro-clusters.
- *Generalizable and domain independent* – we show results on two diverse areas, namely, Twitter data, and HTdata; as well as on multiple languages, i.e. Spanish, Italian, English.

**Reproducibility:** Our code is open-sourced at <https://github.com/mengshih/InfoShield>. The HT dataset is available to researchers after NDA (email Dr. Cara Jones [caraj@manassasvtc.com](mailto:caraj@manassasvtc.com)). The Twitter datasets are publicly available (see [4]).

**II. BACKGROUND AND RELATED WORK**

There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

**A. Human Trafficking Detection**

Some previous works try to classify whether or not a particular advertisement is suspected of HT [5], [6], [7], [8]. For instance, HTDN [7] proposes a supervised deep multiclass model trained on 10K manually labeled ads. Their results are later improved, on the same data, using an off-the-shelf neural network [9]. Unfortunately, due to the adversarial nature of exact advertisements, these predefined or learned features don’t stay relevant over time. These labeled ads are also expensive to obtain (requiring the precious time of domain experts) and are error-prone, as will be discussed in Section V. Moreover, inspecting ads individually, we might overlook ads that are part of an organized activity but do not stand out on their own. Therefore, unsupervised algorithms that find groups of organized activity are preferred in this domain. Template Matching [10] exploits the above insight, being the first anti-HT method to use knowledge to perform clustering. However, the interpretability of clusters is limited, and the algorithm isn’t scalable.

**B. Social Media Bot Detection**

Most efforts in detecting bots in social media platforms are formulated as supervised classification based on features from users and the content they post [11], [12]. Fewer works look for anomalies or fraud in networks, rather than in text, for instance [13]. A notable method, Botometer [14], formerly called BotNois, is an online service that provides a score of likelihood that a particular user is a bot. Since it is the only state-of-the-art method with public access to the implementation, we will use it as a baseline for our experiments in Section V. [4] gives a more comprehensive overview of Twitter bot detection methods, and also provides the dataset we will use in Section V-A1.

Very few works focus on detecting organized activity: groups working together to mislead people about who they are and what they are doing, which is a rising issue [15]. ND-Syn [16] finds a related but different type of behavior, i.e. “retweet spam”, where groups of multiple users exhibit organized behavior by consistently retweeting a particular user’s tweets.

# 'F' – which one?

- Pick one to review
- Q1) Which one?
- Q2) Why?

(a)

(b)

(c)

**C. Document Embedding and Clustering**  
 Much work has been done to represent documents as a machine-understandable format. The most widely-used approaches to represent documents include bag-of-words [17] and term frequency-inverse document frequency (tf-idf) [18]. These methods are commonly used for plagiarism detection [19], [20], [21], [22], which is a similar setting to near-duplicate detection. However, none of these methods do visualization or ranking, and some assumptions do not work in our case, i.e. [22] assumes documents consist of multiple lines, which is not the case for tweets or the majority of exact advertisements.

**A. Application to the Human Trafficking Domain**  
 While INFOSHIELD is general, our main motivation is near-duplicate detection and summarization in exact advertisement, human trafficking (HT) is a dangerous social problem which is difficult to tackle. It is estimated that there are 24.9 million people trapped in forced labor, 55% of which are women and girls accounting for 90% of victims in the commercial sex industry [1]. The majority of victims are abandoned online [2] and 56% of victims have no input on ad content [3]. The average prey has 4.4 victims [3]. Thus, the majority of ads suspected of HT are written by one person, who is contacting ads for 4.6 different victims at a time. By looking for small clusters of ads that contain similar phrasing, rather than analyzing standalone ads, we're finding the groups of ads that are most likely to be organized activity, which is a strong signal of HT.  
 Currently, law enforcement looks for HT cases manually, often one at a time. Our proposed INFOSHIELD will help them save time by detecting micro-clusters of similar ads, grouping them, and summarizing the common parts, as shown in Figure 2, which depicts Twitter data – we refrain from showing exact ad results for the victims' safety.

**I. INTRODUCTION**  
 Given many documents, the majority of which do not belong to any cluster, how can we find small clusters of related documents? The driving application is human trafficking detection, where exact ads that are very similar are usually a sign of trafficking.  
 Finding related documents is a problem with numerous applications, such as search engines, plagiarism detection, mailing address de-duplication, and more.  
 In this paper, we develop INFOSHIELD, a general, information theory based method, and we illustrate its generality, effectiveness and scalability on two settings: tweets. Twitter. This particular application benefits from a vast amount of publicly available data.

\* Both authors contributed equally to this work.

grammatical errors are common. Thus, these methods are not generalizable.  
**B. Minimum Description Length**  
 The Minimum Description Length principle (MDL) [17] assumes that the best model  $M \in \mathcal{M}$  for data  $D$  minimizes  $C(M) = C(D|M)$ , where  $C(\cdot)$  is defined as the cost, i.e. number of bits, needed to describe  $\cdot$  losslessly. The main insight is that it penalizes both the model cost  $C(M)$ , as well as the encoding of encodings from the model  $C(D|M)$  – while several other methods ignore the model complexity, MDL has been extremely successful in several data mining applications.

Dataset	Twitter Data				Human Trafficking Data			
	Dataset	Size	Time	F1	Dataset	Size	Time	F1
Human Trafficking	249	100	21	94%	Twitter	100	100	91%
Exact (E)	100	100	100	100	Exact (E)	100	100	100
Twitter (T)	100	100	28%	85%	Twitter (T)	100	100	85%
InfoShield (I)	100	100	19%	26%	InfoShield (I)	100	100	26%
HT (H)	100	100	19%	26%	HT (H)	100	100	26%

TABLE VIII. INFOSHIELD performs well. Notice that INFOSHIELD beats or approaches the best domain-specific method in both settings. Bold shows the best score, underline shows methods within 10 points of the best. Methods in red are supervised, while INFOSHIELD is unsupervised.

Fig. 1. INFOSHIELD works: being precise (left), scalable (middle), and interpretable (right), detecting and visualizing clusters (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

**C. Our Method**  
 Our first insight is to formalize the problem with information theory, and use the Minimum Description Length (MDL) principle to find good templates, which represent cluster text, with "slots", i.e. parts of the template that differ for each document. We mark slots with red highlights in Figure 1 (right). We then use this summary to visualize the cluster. INFOSHIELD is parameter-free, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the INFOSHIELD-FINE part of our method.  
 The second insight is a novel pre-processing method, INFOSHIELD-COARSE, that dramatically improves scalability to be quasi-linear, by (a) eliminating single-copy documents/ads and (b) grouping the rest in coarse, but mainly homogeneous, clusters.  
 The resulting INFOSHIELD has a long list of desirable properties: It is

- **Practical**, processing 1 million documents within 2 hours on a Dell Alienware laptop with 32GB RAM and 7 processor running Arch Linux, and correctly identifying Twitter spamshots with an F1 score of 90% or higher.
- **Parameter-free & Principled**, requiring no user-defined parameters thanks to the Minimum Description Language principle.
- **Interpretable**, providing a clear visualization and summarization of the discovered micro-clusters.
- **Generalizable and domain independent** – we show results on two diverse areas, namely, Twitter data, and HT data, as well as on multiple languages, i.e. Spanish, Italian, English.

**Reproducibility:** Our code is open-sourced at <https://github.com/mengshih/InfoShield>. The HT dataset is available to researchers after NDA (email: Dr. Cara Jones [carajones@trainsystemstics.com](mailto:carajones@trainsystemstics.com)). The Twitter datasets are publicly available (see [4]).

**II. BACKGROUND AND RELATED WORK**  
 There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

**A. Human Trafficking Detection**  
 Some previous works try to classify whether or not a particular advertisement is suspected of HT [5], [6], [7], [8]. For instance, HTDN [7] proposes a supervised deep multilingual model trained on 10K manually labeled ads. Their results are later improved, on the same data, using an off-the-shelf neural network [9]. Unfortunately, due to the adversarial nature of exact advertisements, these predefined or learned features don't stay relevant over time. These labeled ads are also expensive to obtain (requiring the previous time of domain experts) and are error-prone, as will be discussed in Section V. Moreover, inspecting ads individually, we might overlook ads that are part of an organized activity but do not stand out on their own. Therefore, unsupervised algorithms that find groups of organized activity are preferred in this domain. Template Matching [10] exploits the above insight, being the first anti-HT method to use knowledge to perform clustering. However, the interpretability of clusters is limited, and the algorithm isn't scalable.

**B. Social Media Bot Detection**  
 Most efforts in detecting bots in social media platforms are formulated as supervised classification based on features from users and the content they post [11], [12]. Fewer works look for anomalies or fraud in networks, rather than in text, for instance [13]. A notable method, Botometer [14], formerly called BotNois, is an online service that provides a score of likelihood that a particular user is a bot. Since it is the only state-of-the-art method with public access to the implementation, we will use it as a baseline for our experiments in Section V. [4] gives a more comprehensive overview of Twitter bot detection methods, and also provides the dataset we will use in Section V-A. Very few works focus on detecting organizational activity – groups working together to mislead people about who they are and what they are doing, which is a rising issue [15]. ND-Sync [16] finds a related but different type of behavior, i.e. "retweet spam", where groups of multiple users exhibit coherent behavior by consistently retweeting a particular user's tweets.

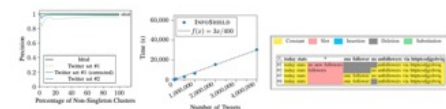


Fig. 1. INFOSHIELD works: being precise (left), scalable (middle), and interpretable (right), detecting and visualizing clusters (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

**C. Our Method**  
 Our first insight is to formalize the problem with information theory, and use the Minimum Description Length (MDL) principle to find good templates, which represent cluster text, with "slots", i.e. parts of the template that differ for each document. We mark slots with red highlights in Figure 1 (right). We then use this summary to visualize the cluster. INFOSHIELD is parameter-free, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the INFOSHIELD-FINE part of our method.  
 The second insight is a novel pre-processing method, INFOSHIELD-COARSE, that dramatically improves scalability to be quasi-linear, by (a) eliminating single-copy documents/ads and (b) grouping the rest in coarse, but mainly homogeneous, clusters.  
 The resulting INFOSHIELD has a long list of desirable properties: It is

- **Practical**, processing 1 million documents within 2 hours on a Dell Alienware laptop with 32GB RAM and 7 processor running Arch Linux, and correctly identifying Twitter spamshots with an F1 score of 90% or higher.
- **Parameter-free & Principled**, requiring no user-defined parameters thanks to the Minimum Description Language principle.
- **Interpretable**, providing a clear visualization and summarization of the discovered micro-clusters.
- **Generalizable and domain independent** – we show results on two diverse areas, namely, Twitter data, and HT data, as well as on multiple languages, i.e. Spanish, Italian, English.

**Reproducibility:** Our code is open-sourced at <https://github.com/mengshih/InfoShield>. The HT dataset is available to researchers after NDA (email: Dr. Cara Jones [carajones@trainsystemstics.com](mailto:carajones@trainsystemstics.com)). The Twitter datasets are publicly available (see [4]).

**II. BACKGROUND AND RELATED WORK**  
 There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

**A. Human Trafficking Detection**  
 Some previous works try to classify whether or not a particular advertisement is suspected of HT [5], [6], [7], [8]. For instance, HTDN [7] proposes a supervised deep multilingual model trained on 10K manually labeled ads. Their results are later improved, on the same data, using an off-the-shelf neural network [9]. Unfortunately, due to the adversarial nature of exact advertisements, these predefined or learned features don't stay relevant over time. These labeled ads are also expensive to obtain (requiring the previous time of domain experts) and are error-prone, as will be discussed in Section V. Moreover, inspecting ads individually, we might overlook ads that are part of an organized activity but do not stand out on their own. Therefore, unsupervised algorithms that find groups of organized activity are preferred in this domain. Template Matching [10] exploits the above insight, being the first anti-HT method to use knowledge to perform clustering. However, the interpretability of clusters is limited, and the algorithm isn't scalable.

**B. Social Media Bot Detection**  
 Most efforts in detecting bots in social media platforms are formulated as supervised classification based on features from users and the content they post [11], [12]. Fewer works look for anomalies or fraud in networks, rather than in text, for instance [13]. A notable method, Botometer [14], formerly called BotNois, is an online service that provides a score of likelihood that a particular user is a bot. Since it is the only state-of-the-art method with public access to the implementation, we will use it as a baseline for our experiments in Section V. [4] gives a more comprehensive overview of Twitter bot detection methods, and also provides the dataset we will use in Section V-A. Very few works focus on detecting organizational activity – groups working together to mislead people about who they are and what they are doing, which is a rising issue [15]. ND-Sync [16] finds a related but different type of behavior, i.e. "retweet spam", where groups of multiple users exhibit coherent behavior by consistently retweeting a particular user's tweets.

# ‘F’ – which one?



- Pick one to review
- Q1) Which one? -> ‘c’
- Q2) Why?

(a)

(b)

(c)



**C. Document Embedding and Clustering**  
 Much work has been done to represent documents in a machine-understandable format. The most widely-used approaches to represent documents include bag of words [1] and term frequency-inverse document frequency (TF-IDF) [18]. These methods are commonly used for plagiarism detection [19], [20], [21], [22], which is a similar setting to near-duplicate detection. However, none of these methods do visualization or ranking, and some assumptions do not work in our case, i.e. [22] assumes documents consist of multiple lines, which is not the case for tweets or the majority of exact advertisements.

**Human Trafficking Detection**  
 Some previous works try to classify whether or not a particular advertisement is suspected of HT [5], [6], [7], [8]. For instance, HTDN [7] proposes a supervised deep multilingual model trained on 10K manually labeled ads. Their results are later improved, on the same data, using an off-the-shelf neural network [9]. Unfortunately, due to the adversarial nature of exact advertisements, these predefined or learned features don't stay relevant over time. These labeled ads are also expensive to obtain (requiring the previous time of domain experts) and are error-prone, as will be discussed in Section V. Moreover, inspecting ads individually, we might overlook ads that are part of an organized activity but do not stand out on their own. Therefore, unsupervised algorithms that find groups of organized activity are preferred in this domain. Template Matching [10] exploits the above insight, being the first anti-HT method to our knowledge to perform clustering. However, the interpretability of clusters is limited, and the algorithm isn't scalable.

**Introduction**  
 Given many documents, the majority of which do not belong to any cluster, how can we find small clusters of related documents? The driving application is human trafficking detection, where exact ads that are very similar are usually a sign of trafficking. Finding related documents is a problem with numerous applications, such as search engines, plagiarism detection, mailing address de-duplication, and more.  
 In this paper, we develop INFOSHIELD, a general, information theory based method, and we illustrate its generality, effectiveness and scalability on two settings: tweets and Twitter. This particular application benefits from a vast amount of publicly available data.

\* Both authors contributed equally to this work.

grammatical errors are common. Thus, these methods are not generalizable.  
**Minimum Description Length**  
 The Minimum Description Length principle (MDL) [17] assumes that the best model  $M \in \mathcal{M}$  for data  $D$  minimizes  $C(M) = C(D|M)$ , where  $C(\cdot)$  is defined as the cost, i.e. number of bits, needed to describe  $\cdot$  losslessly. The main insight is that it penalizes both the model cost  $C(M)$ , as well as the encoding of encodings from the model  $C(D|M)$  - while several other methods ignore the model complexity. MDL has been extremely successful in several data mining applications.

Dataset	Twitter Data				Human Trafficking Data			
	Dataset	Size	Time	Time	Dataset	Size	Time	Time
Twitter	240	100	11	240	Twitter	100	11	240
Human Trafficking	100	100	11	100	Human Trafficking	100	11	100
Exact (E)	100	100	11	100	Exact (E)	100	11	100
Similar (S)	100	100	11	100	Similar (S)	100	11	100
Total (T)	100	100	11	100	Total (T)	100	11	100

TABLE VIII. INFOSHIELD performs well. Notice that INFOSHIELD beats or approaches the best domain-specific method in both settings. Bold shows the best score, underline shows methods within 10 points of the best. Methods in red are supervised, while INFOSHIELD is unsupervised.

Fig. 1. INFOSHIELD works: being precise (left), scalable (middle), and interpretable (right), detecting and visualizing clusters (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

**C. Our Method**  
 Our first insight is to formalize the problem with information theory, and use the Minimum Description Length (MDL) principle to find good templates, which represent cluster text, with “holes”, i.e. parts of the template that differ for each document. We mark slots with red highlights in Figure 1 (right). We then use this summary to visualize the cluster. INFOSHIELD is *parameter-free*, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the INFOSHIELD-FIT part of our method. The second insight is a novel pre-processing method, INFOSHIELD-COARSE, that dramatically improves scalability to be quasi-linear, by (a) eliminating single-copy documents/ads and (b) grouping the rest in coarse, but mainly homogeneous, clusters. The resulting INFOSHIELD has a long list of desirable properties: It is

- *Practical*, processing 1 million documents within 2 hours on a Dell Alienware laptop with 32GB RAM and i7 processor running Arch Linux, and correctly identifying Twitter spamshots with an F1 score of 90% or higher.
- *Parameter-free & Principled*, requiring no user-defined parameters thanks to the Minimum Description Language principle.
- *Interpretable*, providing a clear visualization and summarization of the discovered micro-clusters.
- *Generalizable and domain independent* - we show results on two diverse areas, namely, Twitter data, and HT/ads; as well as on multiple languages, i.e. Spanish, Italian, English.

**Reproducibility:** Our code is open-sourced at <https://github.com/rennshilke/InfoShield>. The HT dataset is available to researchers after NDA (email Dr. Cara Jones [carajones@cmu.edu](mailto:carajones@cmu.edu)). The Twitter datasets are publicly available (see [4]).

**II. BACKGROUND AND RELATED WORK**  
 There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

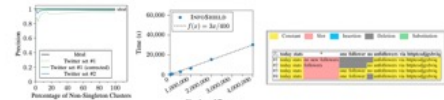


Fig. 1. INFOSHIELD works: being precise (left), scalable (middle), and interpretable (right), detecting and visualizing clusters (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

**C. Our Method**  
 Our first insight is to formalize the problem with information theory, and use the Minimum Description Length (MDL) principle to find good templates, which represent cluster text, with “holes”, i.e. parts of the template that differ for each document. We mark slots with red highlights in Figure 1 (right). We then use this summary to visualize the cluster. INFOSHIELD is *parameter-free*, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the INFOSHIELD-FIT part of our method. The second insight is a novel pre-processing method, INFOSHIELD-COARSE, that dramatically improves scalability to be quasi-linear, by (a) eliminating single-copy documents/ads and (b) grouping the rest in coarse, but mainly homogeneous, clusters. The resulting INFOSHIELD has a long list of desirable properties: It is

- *Practical*, processing 1 million documents within 2 hours on a Dell Alienware laptop with 32GB RAM and i7 processor running Arch Linux, and correctly identifying Twitter spamshots with an F1 score of 90% or higher.
- *Parameter-free & Principled*, requiring no user-defined parameters thanks to the Minimum Description Language principle.
- *Interpretable*, providing a clear visualization and summarization of the discovered micro-clusters.
- *Generalizable and domain independent* - we show results on two diverse areas, namely, Twitter data, and HT/ads; as well as on multiple languages, i.e. Spanish, Italian, English.

**Reproducibility:** Our code is open-sourced at <https://github.com/rennshilke/InfoShield>. The HT dataset is available to researchers after NDA (email Dr. Cara Jones [carajones@cmu.edu](mailto:carajones@cmu.edu)). The Twitter datasets are publicly available (see [4]).

**II. BACKGROUND AND RELATED WORK**  
 There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

# ‘F’ – which one?

- Pick one to review
- Q1) Which one? -> ‘c’
- Q2) Why?

- A2.1: ‘an image is worth a thousand words’
- A2.2: the author respects reviewer’s time

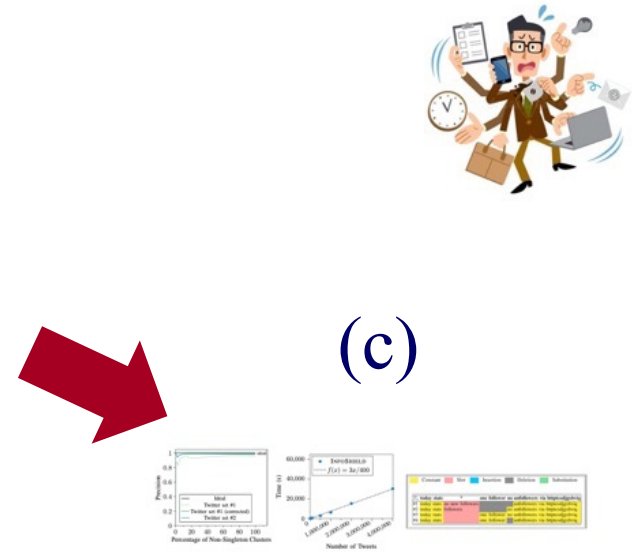


Fig. 1. INFOSHIELD works: being precise (left), scalable (middle), and interpretable (right), detecting and visualizing slots (in red), i.e. portions of tweets that highly differ between otherwise duplicate documents.

#### C. Our Method

Our first insight is to formalize the problem with information theory, and use the Minimum Description Length (MDL) principle to find good templates, which represent cluster text, with “slots”, i.e. parts of the template that differ for each document. We mark slots with red highlights in Figure 1 (right). We then use this summary to visualize the cluster. INFOSHIELD is parameter-free, since MDL can automatically pick the best choice of parameter values for any algorithm by choosing the combination with the shortest compression length. This is the INFOSHIELD-core part of our method.

The second insight is a novel pre-processing method, INFOSHIELD-coarse, that dramatically improves scalability to be quasi-linear, by (a) eliminating single-copy documents and (b) grouping the rest in coarse, but mainly homogeneous, clusters.

The resulting INFOSHIELD has a long list of desirable properties. It is

- *Practical*, processing 1 million documents within 2 hours on a Dell Alienware laptop with 32GB RAM and i7 processor running Arch Linux; and correctly identifying Twitter accounts with an F1 score of 90% or higher.
- *Parameter-free & Principled*, requiring no user-defined parameters thanks to the Minimum Description Language principle.
- *Interpretable*, providing a clear visualization and summarization of the discovered micro-clusters.
- *Generalizable and domain independent* – we show results on two diverse areas, namely, Twitter data, and HTData; as well as on multiple languages, i.e. Spanish, Italian, English.

**Reproducibility:** Our code is open-sourced at <https://github.com/robertoschiffano/InfoShield>. The HT dataset is available to researchers after NDA (email Dr. Cara Jones [caraj@twitteranalytics.com](mailto:caraj@twitteranalytics.com)). The Twitter datasets are publicly available (see [4]).

#### II. BACKGROUND AND RELATED WORK

There is a lot of work on HT detection, document clustering, and multiple sequence alignment, and we group it in the following sub-sections.

#### A. Human Trafficking Detection

Some previous works try to classify whether or not a particular advertisement is suspected of HT [5], [6], [7], [8]. For instance, HTEN [7] proposes a supervised deep multistage model trained on 10K manually labeled ads. Their results are later improved, on the same data, using an ordinal regression neural network [9]. Unfortunately, due to the adversarial nature of evict advertisements, these predefined or learned features don’t stay relevant over time. These labeled ads are also expensive to obtain (requiring the precious time of domain experts) and are error-prone, as will be discussed in Section V. Moreover, inspecting ads individually, we might overlook ads that are part of an organized activity but do not stand out on their own. Therefore, unsupervised algorithms that find groups of organized activity are preferred in this domain. Template Matching [10] exploits the above insight, being the first anti-HT method to our knowledge to perform clustering. However, the interpretability of clusters is limited, and the algorithm isn’t scalable.

#### B. Social Media Bot Detection

Most efforts in detecting bots in social media platforms are formulated as supervised classification based on features from users and the content they post [11], [12]. Fewer works look for anomalies or fraud in networks, rather than in text, for instance [13]. A notable method, Botometer [14], formerly called BotvNol, is an online service that provides a score of likelihood that a particular user is a bot. Since it is the only state-of-the-art method with public access to the implementation, we will use it as a baseline for our experiments in Section V. [6] gives a more comprehensive overview of Twitter bot detection methods, and also provides the dataset we will use in Section V-C. Very few works focus on detecting organized activity – groups working together to mislead people about who they are and what they are doing, which is a rising issue [15]. ND-Sync [16] finds a related but different type of behavior, i.e. “retweet spars”, where groups of multiple users exhibit organized behavior by consistently retweeting a particular user’s tweets.

# Outline



- Top 3 lessons: ‘F.A.N.’



- **F**igure (‘crown jewel’ figure)
- A
- N



- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions

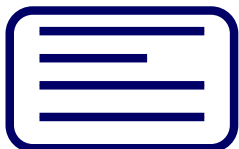


# 'F': What makes a good 'crown jewel' figure?

- [Consider a paper that tries to find near-duplicate escort ads -> human trafficking ]



...



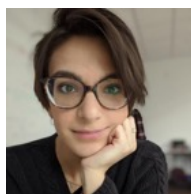


(paper)

Meng-Chieh Lee, Catalina Vajiac, Aayushi Kulshrestha, Sacha Levy, Namyong Park, Cara Jones, Reihaneh Rabbany, and Christos Faloutsos

*INFOSHIELD: Generalizable Information-Theoretic Human-Trafficking Detection*

ICDE 2021, Chania, Greece, April 2021.



# 'F': What makes a good 'crown jewel' figure?

- [Consider a paper that tries to find near-duplicate escort ads -> human trafficking ]

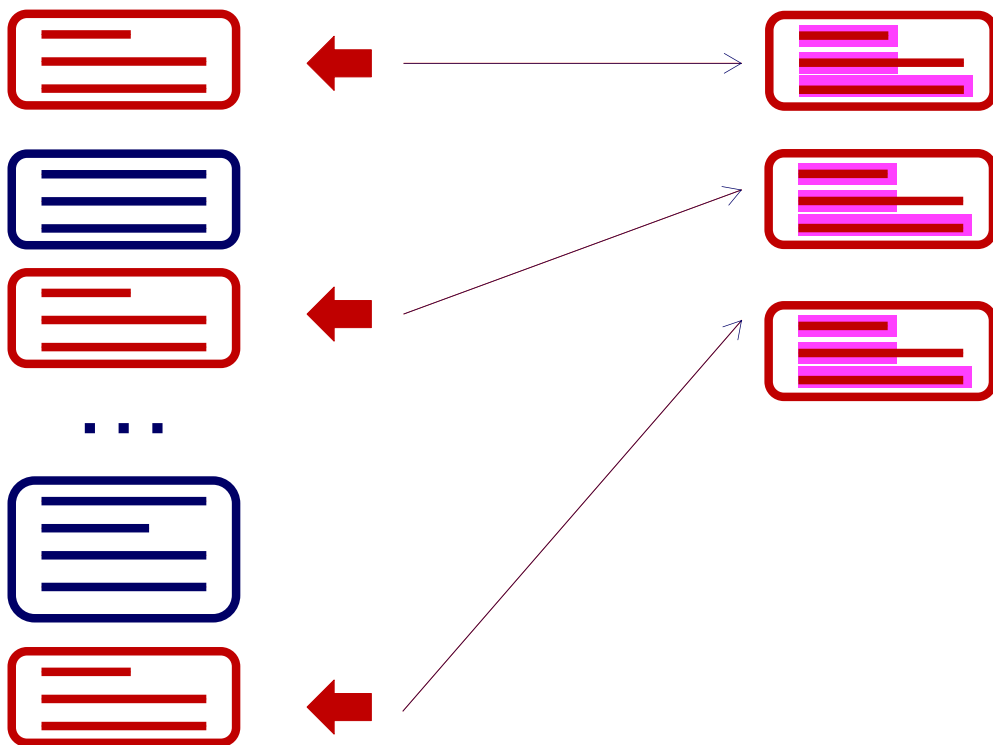


...



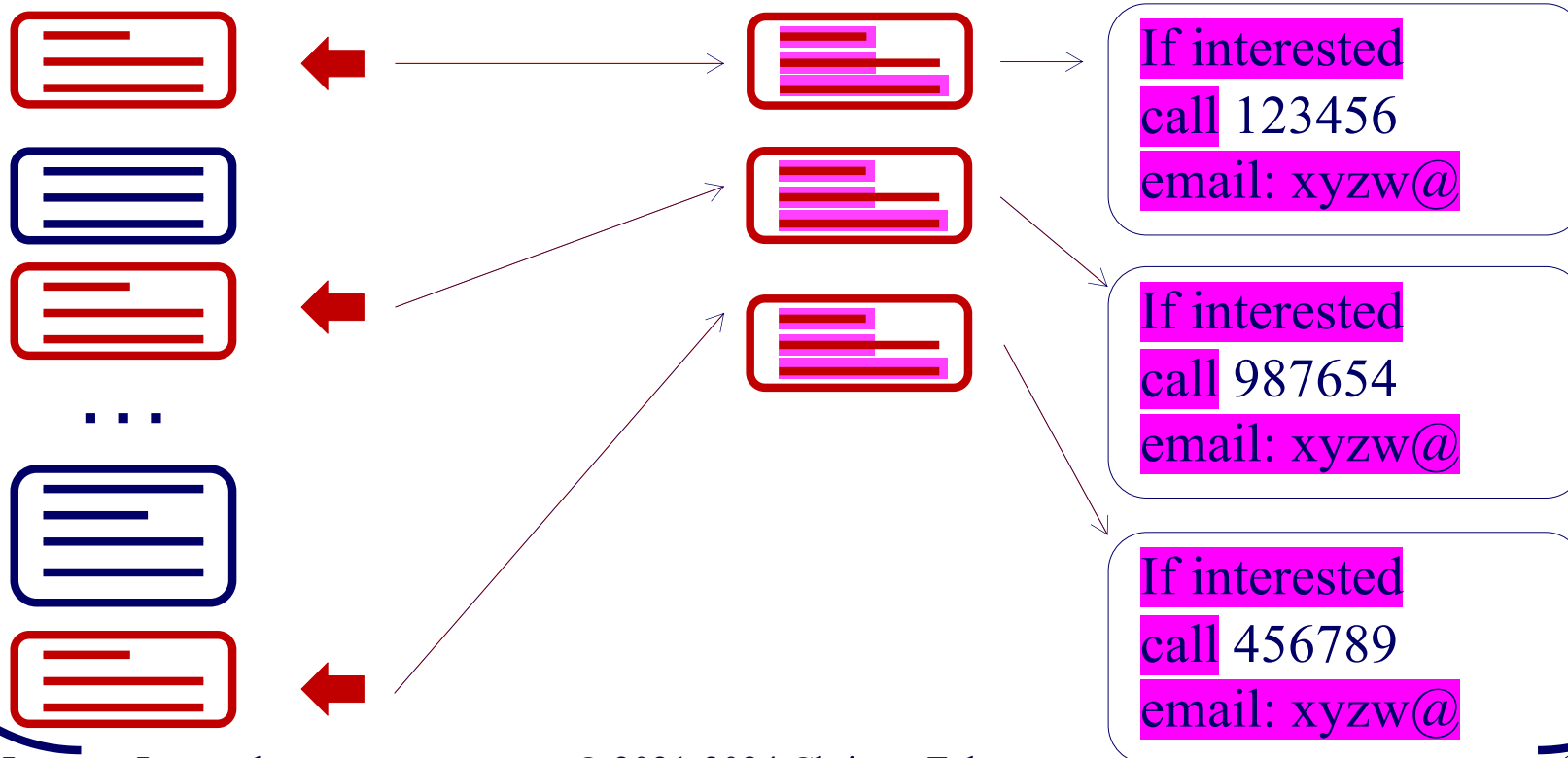
# 'F': What makes a good 'crown jewel' figure?

- [Consider a paper that tries to find near-duplicate escort ads -> human trafficking ]



# 'F': What makes a good 'crown jewel' figure?

- [Consider a paper that tries to find near-duplicate escort ads -> human trafficking ]



# 'F': What makes a good 'crown jewel' figure?

- [Consider a paper that tries to find near-duplicate escort ads -> human trafficking ]



...



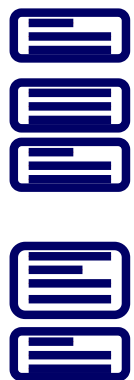
**Mock-up text,  
for victims' safety**

If interested  
call 123456  
email: xyzw@

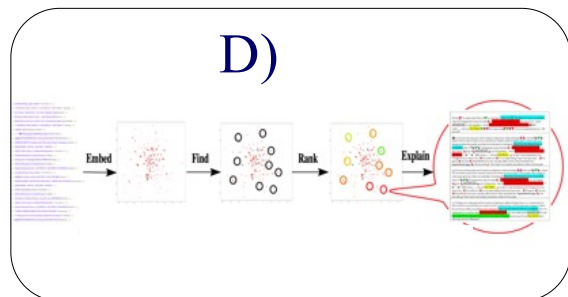
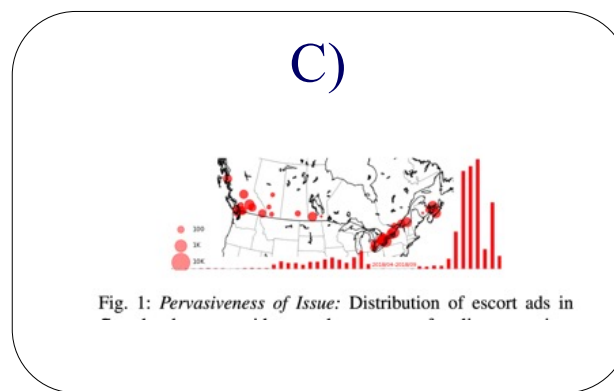
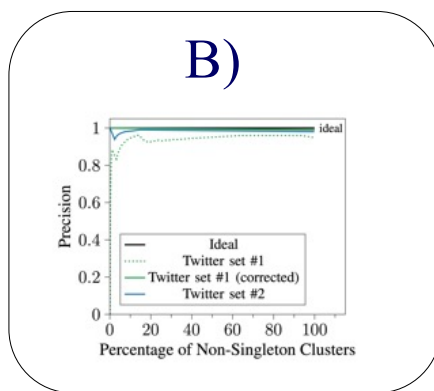
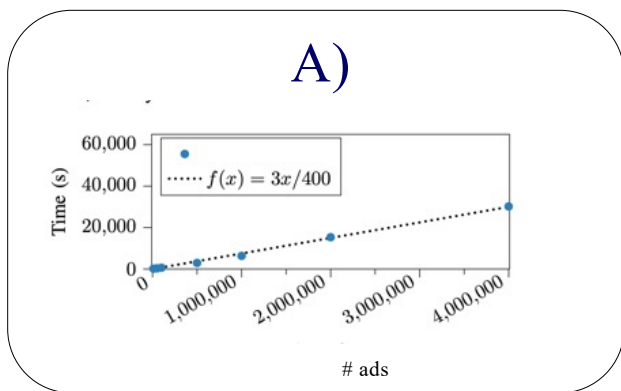
If interested  
call 987654  
email: xyzw@

If interested  
call 456789  
email: xyzw@

# 'F': What makes a good 'crown jewel' figure?



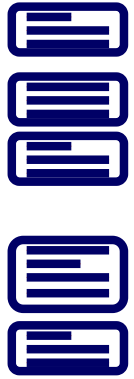
- Which one a reviewer will find most impressive?



E)

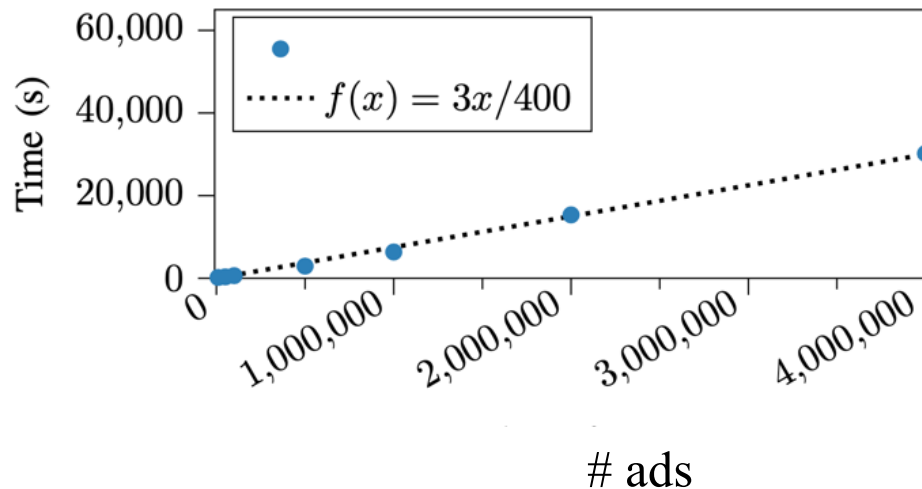
	Constant	Slot	Insertion	Deletion	Substitution
$T_1$ today stats	*	one follower	no unfollowers	via httpcodjgxbviq	
#1 today stats		no new followers		unfollowers via httpcodjgxbviq	
#2 today stats		followers		no unfollowers via httpcodjgxbviq	
#3 today stats			one follower	no unfollowers via httpcodjgxbviq	
#4 today stats			one follower	unfollowers via httpcodjgxbviq	

# 'F': What makes a good 'crown jewel' figure?



- Which one a reviewer will find most impressive?

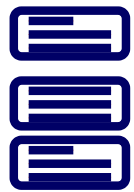
(A)



Message??

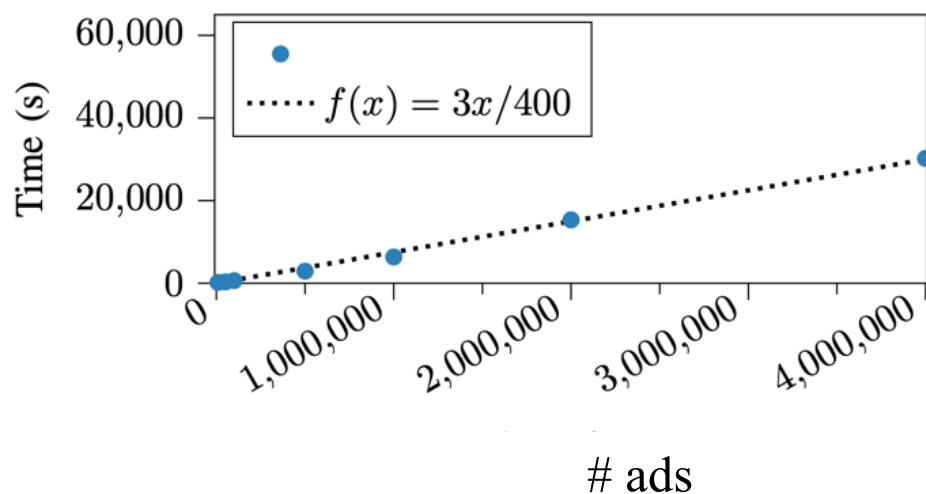


# 'F': What makes a good 'crown jewel' figure?



- Which one a reviewer will find most impressive?

(A)

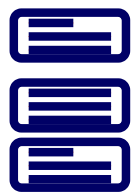


scalable



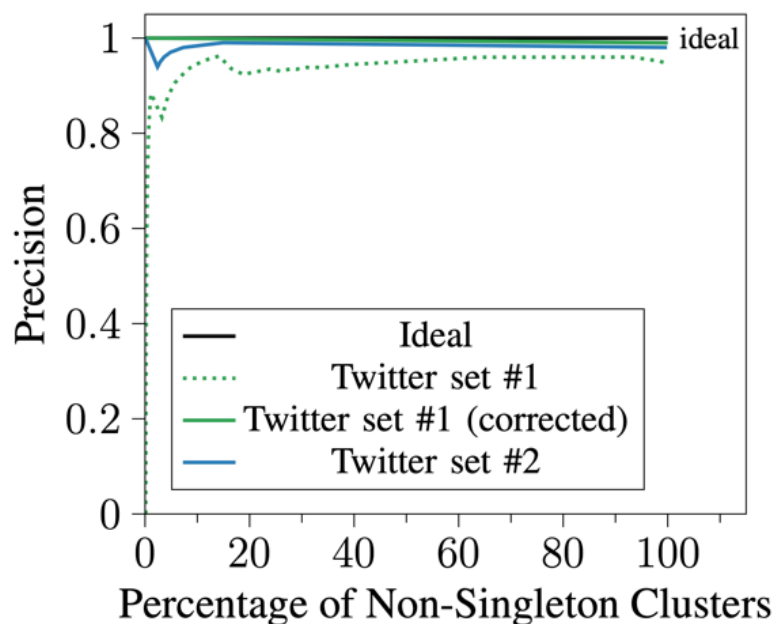


# 'F': What makes a good 'crown jewel' figure?



- Which one a reviewer will find most impressive?

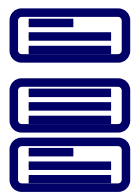
(B)



Message?

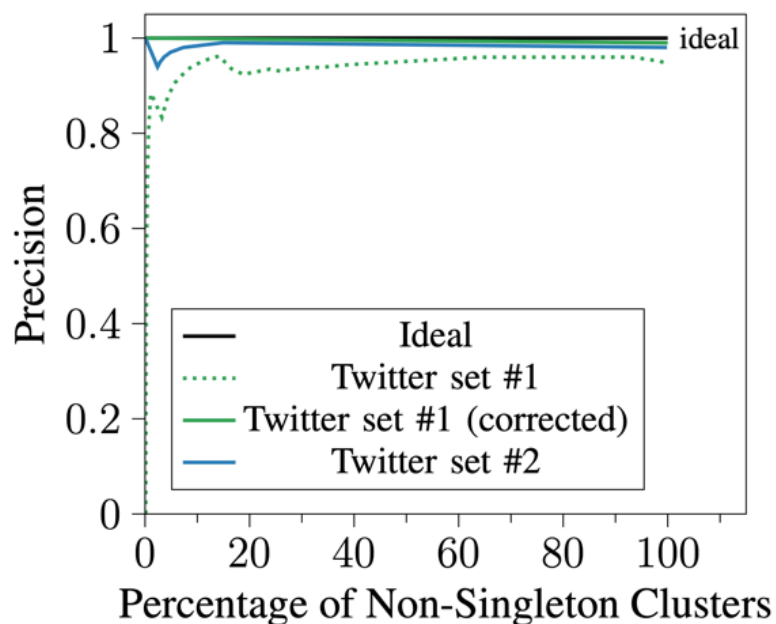


# 'F': What makes a good 'crown jewel' figure?



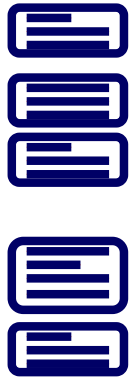
- Which one a reviewer will find most impressive?

(B)



accurate

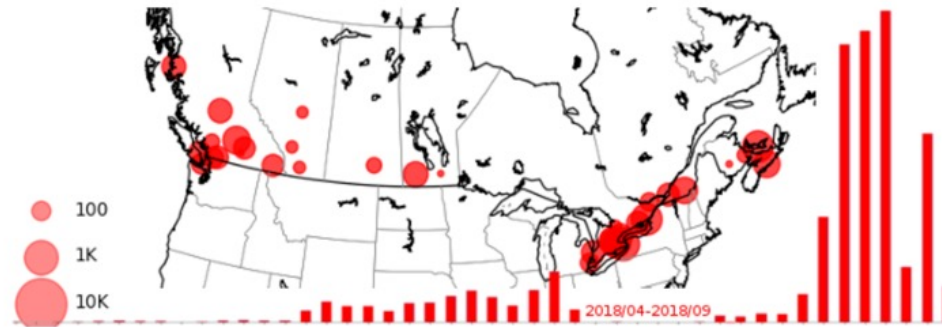




# 'F': What makes a good 'crown jewel' figure?

- Which one a reviewer will find most impressive?

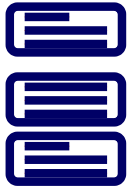
(C)



Message?



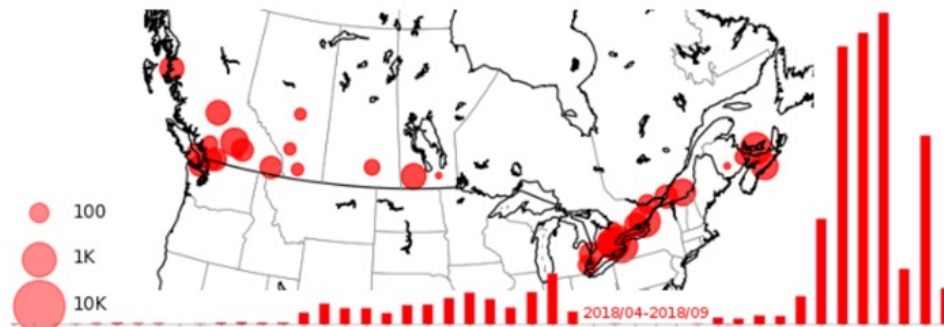
Fig. 1: *Pervasiveness of Issue*: Distribution of escort ads in



# 'F': What makes a good 'crown jewel' figure?

- Which one a reviewer will find most impressive?

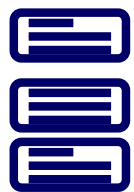
(C)



Important problem



Fig. 1: *Pervasiveness of Issue*: Distribution of escort ads in



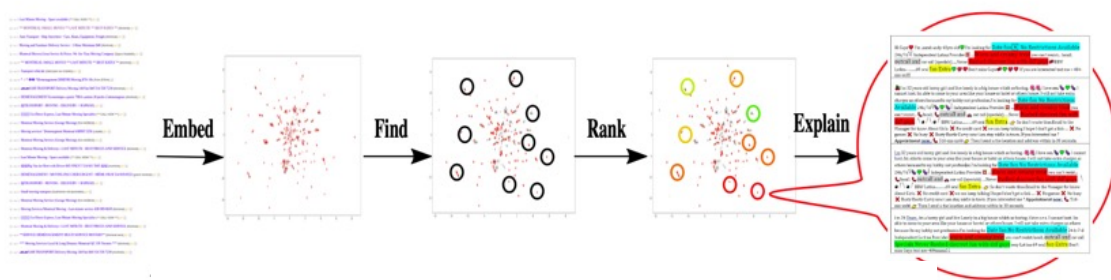
# 'F': What makes a good 'crown jewel' figure?



- Which one a reviewer will find most impressive?

Message?

(D)



System Overview

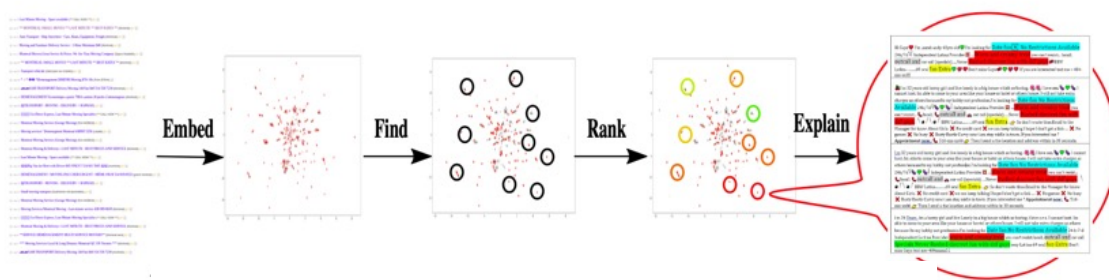


# 'F': What makes a good 'crown jewel' figure?



- Which one a reviewer will find most impressive?

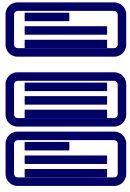
(D)



System Overview

Elaborate  
method





# 'F': What makes a good 'crown jewel' figure?



- Which one a reviewer will find most impressive?

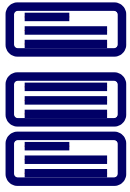
(E)

Message?



	Constant	Slot	Insertion	Deletion	Substitution
$T_1$	today stats	*	one follower	no unfollowers	via httpcodjgxbviq
#1	today stats	no new followers			unfollowers via httpcodjgxbviq
#2	today stats	followers			no unfollowers via httpcodjgxbviq
#3	today stats		one follower		no unfollowers via httpcodjgxbviq
#4	today stats		one follower		unfollowers via httpcodjgxbviq

Result, on real 'tweets'



# 'F': What makes a good 'crown jewel' figure?



- Which one a reviewer will find most impressive?

(E)

It works.

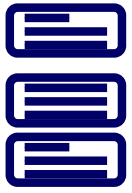


	Constant	Slot	Insertion	Deletion	Substitution
$T_1$	today stats	*	one follower	no unfollowers	via httpcodjgxbviq
#1	today stats	no new followers			unfollowers via httpcodjgxbviq
#2	today stats	followers			no unfollowers via httpcodjgxbviq
#3	today stats		one follower		no unfollowers via httpcodjgxbviq
#4	today stats		one follower		unfollowers via httpcodjgxbviq

Result, on real 'tweets'

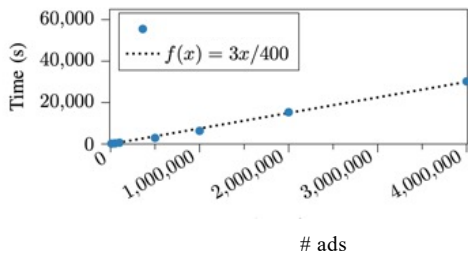


# 'F': What makes a good 'crown jewel' figure?

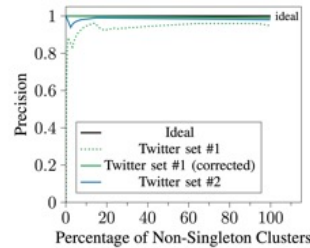


- Which one a reviewer will find most impressive?

A) scalable



B) accurate

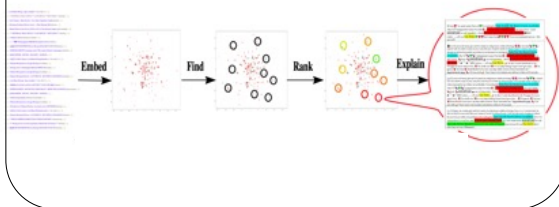


C) important



Fig. 1: Pervasiveness of Issue: Distribution of escort ads in

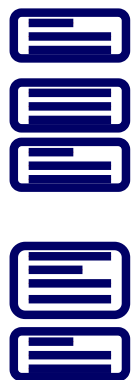
D) elaborate



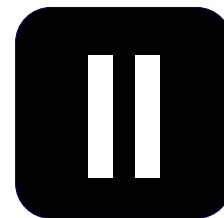
E) It works

	Constant	Slot	Insertion	Deletion	Substitution
$T_1$ today stats	*	one follower	no unfollowers	via httpcodjgxbviq	
#1 today stats	no new followers		unfollowers	via httpcodjgxbviq	
#2 today stats	followers		no unfollowers	via httpcodjgxbviq	
#3 today stats		one follower	no unfollowers	via httpcodjgxbviq	
#4 today stats		one follower	unfollowers	via httpcodjgxbviq	

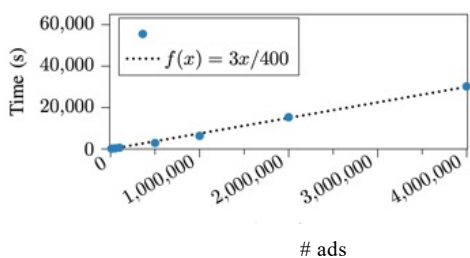
# 'F': What makes a good 'crown jewel' figure?



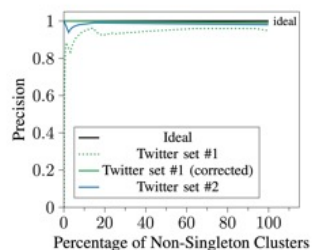
- Which one a reviewer will find most impressive?



A) scalable



B) accurate

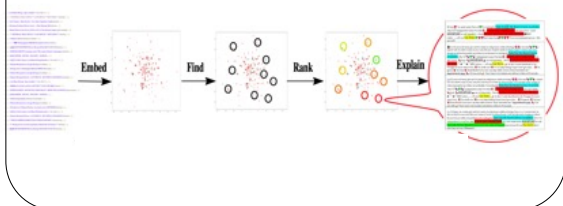


C) important

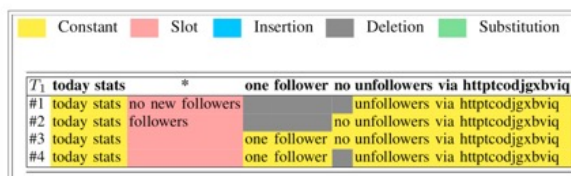


Fig. 1: Pervasiveness of Issue: Distribution of escort ads in

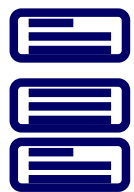
D) elaborate



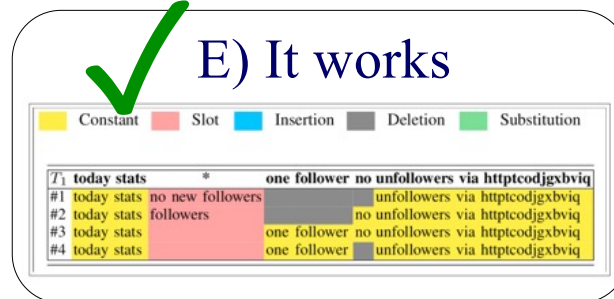
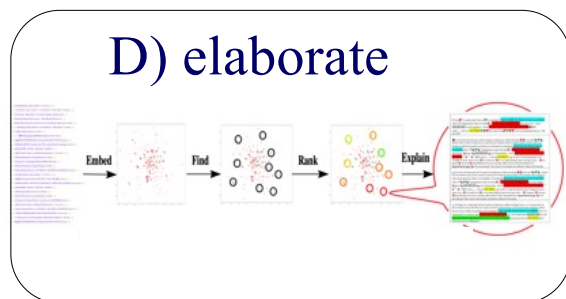
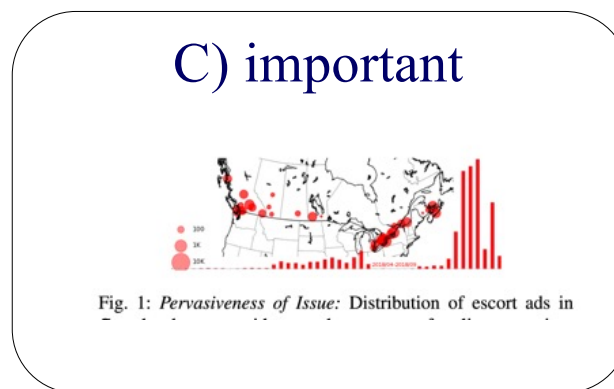
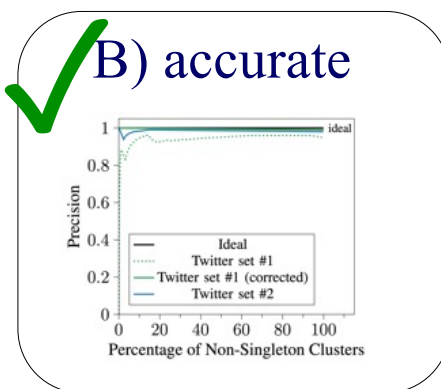
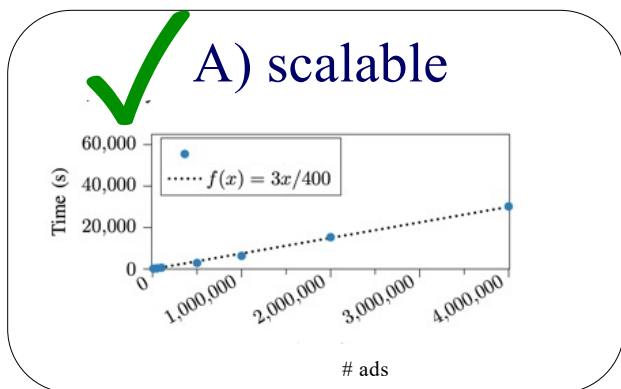
E) It works



# 'F': What makes a good 'crown jewel' figure?



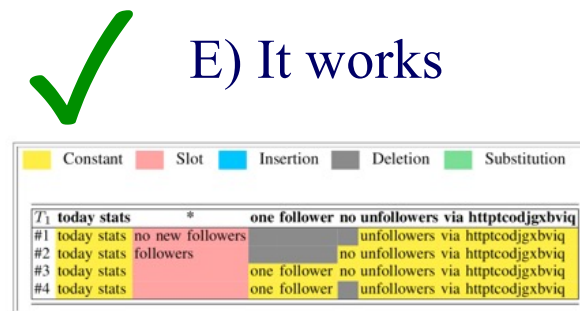
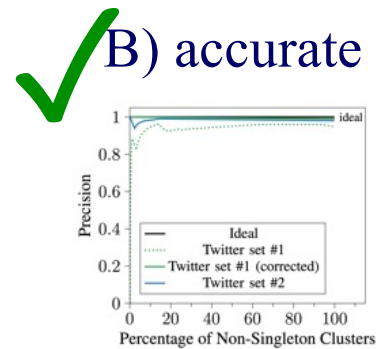
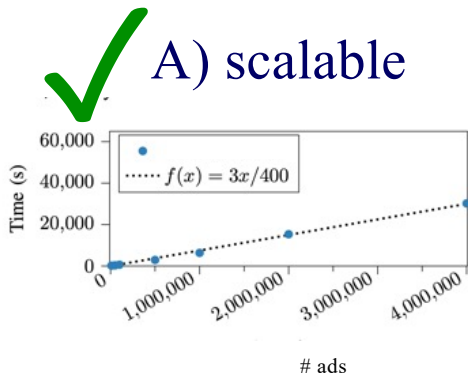
- Q: Which one to choose?



# 'F': What makes a good 'crown jewel' figure?



- A: all 3 - it's OK to have  $>1$  figures





# 'F': What makes a good 'crown jewel' figure?

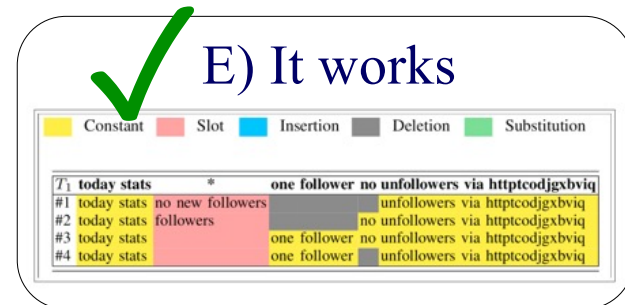
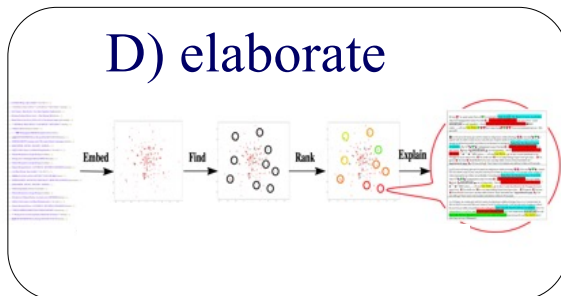


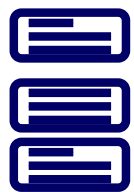
- Emphasize '*what*' we can achieve – not '*how*'



'How'

'What'





# 'F': What makes a good 'crown jewel' figure?

- Emphasize '*what*' we can achieve – not '*how*'



'How'

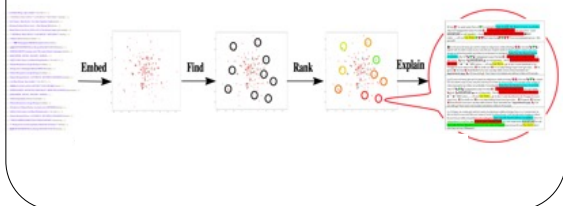
Embed -> cluster -> rank"

“*\*anybody\* could do that – what's so novel?*”

'What'



D) elaborate



E) It works

	Constant	Slot	Insertion	Deletion	Substitution
T <sub>1</sub> today stats	*	one follower	no unfollowers	via httpcodjgxbviq	
#1 today stats	no new followers		unfollowers	via httpcodjgxbviq	
#2 today stats	followers		no unfollowers	via httpcodjgxbviq	
#3 today stats		one follower	no unfollowers	via httpcodjgxbviq	
#4 today stats		one follower	unfollowers	via httpcodjgxbviq	



# 'F': What makes a good 'crown jewel' figure?



- Emphasize '*what*' we can achieve – not '*how*'

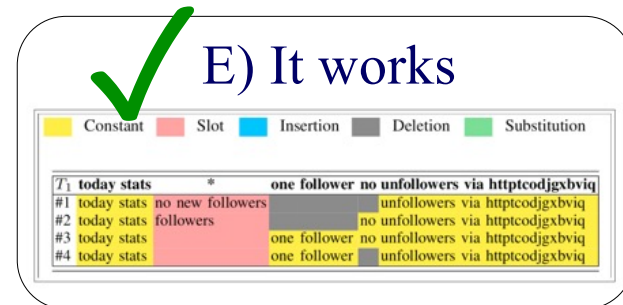
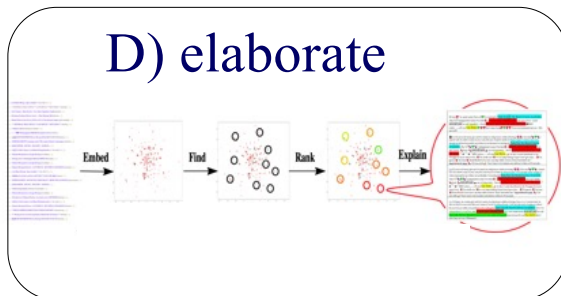


Embed  
 “\*anybody  
 who



→ rank”  
 do that –  
 l?”

‘What’





# 'F': What makes a good 'crown jewel' figure?



- Emphasize '*what*' we can achieve – not '*how*'



Emb

“\*any

”



rank”

that –

”

‘What’



✓ E) It works

	Constant	Slot	Insertion	Deletion	Substitution
T <sub>1</sub> today stats	*	one follower	no unfollowers	via httpcodjgxbviq	
#1 today stats	no new followers			unfollowers via httpcodjgxbviq	
#2 today stats	followers			no unfollowers via httpcodjgxbviq	
#3 today stats		one follower	no unfollowers	via httpcodjgxbviq	
#4 today stats		one follower		unfollowers via httpcodjgxbviq	



# Outline

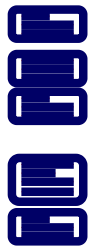


- Top 3 lessons: ‘F.A.N.’
    - Figure (‘crown jewel’ figure)
- ➔
- A
  - N



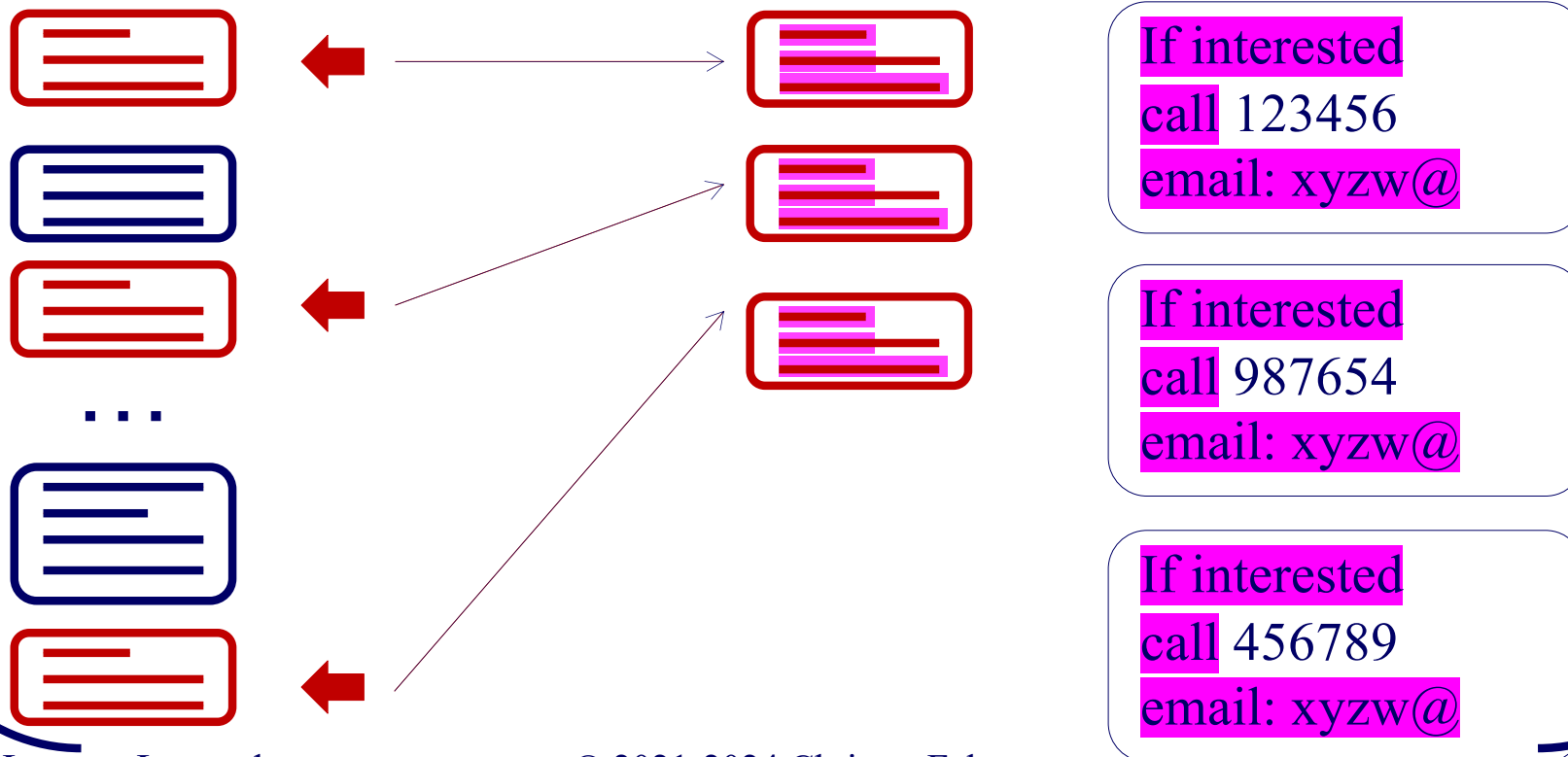
- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions

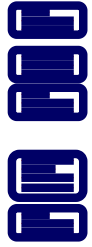




# 'A': Best way to start the abstract?

- [Consider a paper that tries to find near-duplicate escort ads -> human trafficking ]



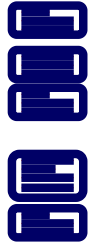


# 'A': Best way to start the abstract?

A)

Human trafficking is an age old problem that continues to affect 25 million people worldwide.

.....



# 'A': Best way to start the abstract?

A)

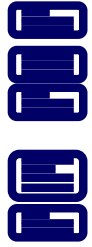
Human trafficking is an age old problem that continues to affect 25 million people worldwide.

.....

B)

Given a million escort advertisements, how can we spot near- duplicates?

.....



# 'A': Best way to start the abstract?

A)

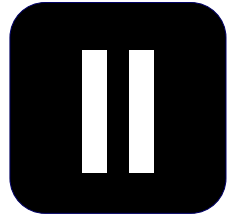
Human trafficking is an age old problem that continues to affect 25 million people worldwide.

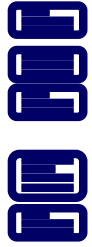
.....

B)

Given a million escort advertisements, how can we spot near- duplicates?

.....





# 'A': Best way to start the abstract?

✓ A)

Human trafficking is an age old problem that continues to affect 25 million people worldwide.

.....

✓✓ B)

Given a million escort advertisements, how can we spot near- duplicates?

.....

# Outline



- Top 3 lessons: ‘F.A.N.’
  - **F**igure (‘crown jewel’ figure)
  - – **A**sk (ask a rhetorical question)
  - N



- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions



# 'A': Best way to start the abstract?

A)

Human trafficking is an age old problem that continues to affect 25 million people worldwide.

.....

B)

Given a million escort advertisements, how can we spot near- duplicates?

.....



‘Rhetorical question’:  
= you know the answer



# 'A': Best way to start the abstract?

A)

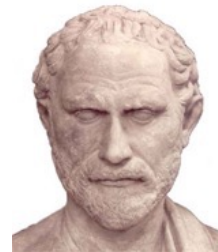
Human trafficking is an age old problem that continues to affect 25 million people worldwide.

.....

B)

Given a million escort advertisements, how can we spot near- duplicates?

.....

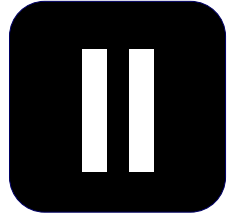


rhetor  
orator



‘Rhetorical question’:  
= you know the answer

# 'A': Best 'rhetorical question':



C)

How can we help law enforcement fight human trafficking?

.....

B)

Given a million escort advertisements, how can we spot near-duplicates?

.....



'Rhetorical question':  
= you know the answer

# 'A': Best 'rhetorical question':

C)

How can we help law enforcement fight human trafficking?

.....



Wrong question.  
(‘call your congressperson’,  
‘deploy face recognition s/w’,  
...)

B)

Given a million escort advertisements, how can we spot near- duplicates?

.....



‘Rhetorical question’:  
= you know the answer

# Outline

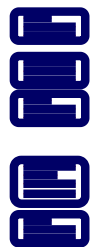


- Top 3 lessons: ‘F.A.N.’
  - **F**igure (‘crown jewel’ figure)
  - **A**sk (ask a rhetorical question)
  - **N**



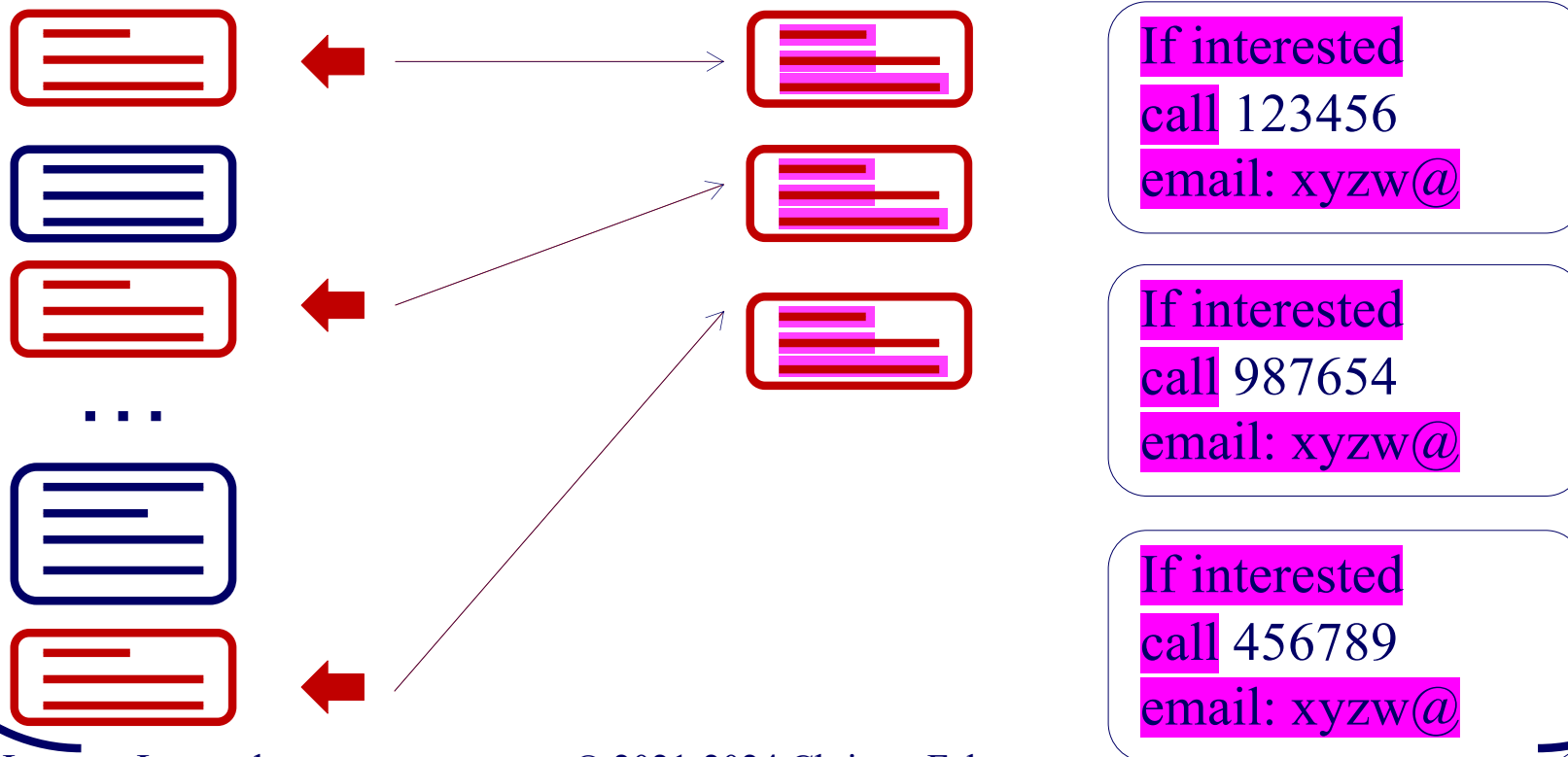
- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions

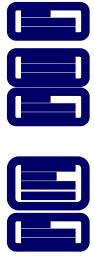




# Reminder: our sample problem

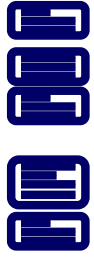
- [Consider a paper that tries to find near-duplicate escort ads -> human trafficking ]





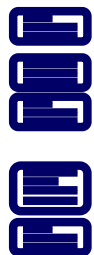
## 'N': Which title is best?

- A. On human trafficking detection
- B. Embedding and clustering for human trafficking detection
- C. Fast and accurate human trafficking detection
- D. TrafficSpot: Fast and accurate human trafficking detection



## 'N': Which title is best?

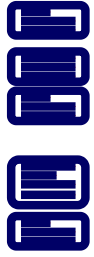
- A. On human trafficking detection
- B. Embedding and clustering for human trafficking detection
- C. Fast and accurate human trafficking detection
- D. TrafficSpot: Fast and accurate human trafficking detection



## 'N': Which title is best?

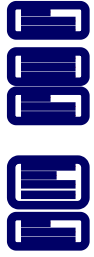
- ~ A. On human trafficking detection *Too general*
- B. Embedding and clustering for human trafficking detection
- C. Fast and accurate human trafficking detection
- D. TrafficSpot: Fast and accurate human trafficking detection





## 'N': Which title is best?

- A. On human trafficking detection
- B. Embedding and clustering for human trafficking detection**
- C. Fast and accurate human trafficking detection
- D. TrafficSpot: Fast and accurate human trafficking detection



# 'N': Which title is best?

A. On human trafficking detection

*NO-NO  
'What' – not 'How'*

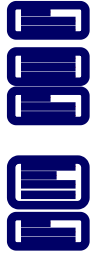


B. Embedding and clustering for human trafficking detection



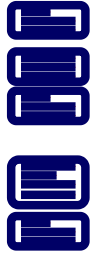
C. Fast and accurate human trafficking detection

D. TrafficSpot: Fast and accurate human trafficking detection



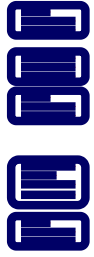
## 'N': Which title is best?

- A. On human trafficking detection
- B. Embedding and clustering for human trafficking detection
- C. Fast and accurate human trafficking detection**
- D. TrafficSpot: Fast and accurate human trafficking detection



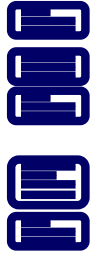
## 'N': Which title is best?

- A. On human trafficking detection
- B. Embedding and clustering for human trafficking detection
- ✓ C. Fast and accurate human trafficking detection
- D. TrafficSpot: Fast and accurate human trafficking detection



## 'N': Which title is best?

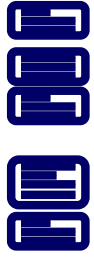
- A. On human trafficking detection
- B. Embedding and clustering for human trafficking detection
- C. Fast and accurate human trafficking detection
- D. TrafficSpot: Fast and accurate human trafficking detection**



## 'N': Which title is best?

- A. On human trafficking detection
- B. Embedding and clustering for human trafficking detection
- C. Fast and accurate human trafficking detection
- ✓✓ D. TrafficSpot: Fast and accurate human trafficking detection



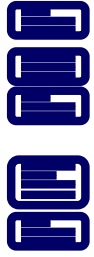


**'N': Which title is best?**

**Q: Benefits of a name?**



✓✓ **D. TrafficSpot: Fast and accurate human trafficking detection**



## 'N': Which title is best?

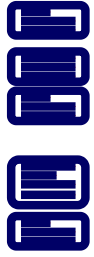
Q: Benefits of a name?

- Easy reference
- .



✓✓ D. TrafficSpot: Fast and accurate human trafficking detection





## 'N': Which title is best?

Q: Benefits of a name?

- Easy reference
- 'novelty'



✓✓ D. TrafficSpot: Fast and accurate human trafficking detection

# Outline

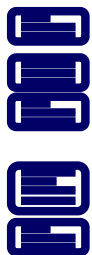


- Top 3 lessons: ‘F.A.N.’
  - **F**igure (‘crown jewel’ figure)
  - **A**sk (ask a rhetorical question)
  - **N**ame (give a name to your method/system)



- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions





# 'N': How to pick good names?



- a) CAE (-> Cluster And Embed)
- b) CUBE (-> **C**l**U**ster and em**B**E**d**)
- c) Spot
- d) TrafficSpot
- e) InfoShield

# 'N': How to pick good names?




a) CAE (-> Cluster And Embed)

b) CUBE (-> **Cl**Uster and em**BE**d)

c) Spot

d) TrafficSpot

e) InfoShield

- Meaningless
-  'how'



# 'N': How to pick good names?



☠ a) CAE (-> Cluster And Embed)

b) CUBE (-> **ClU**ster and em**BE**d)

c) Spot

d) TrafficSpot

e) InfoShield

# 'N': How to pick good names?

☠ a) CAE (-> Cluster And Embed)

☠ b) CUBE (-> **ClU**ster and em**BE**d)

c) Spot

d) TrafficSpot

e) InfoShield

- English word
- irrelevant
- ☠ 'how'



# 'N': How to pick good names?



- ☠ a) CAE (-> Cluster And Embed)
- ☠ b) CUBE (-> **ClU**ster and em**BE**d)
- c) Spot
- d) TrafficSpot
- e) InfoShield

# 'N': How to pick good names?



☠ a) CAE (-> Cluster And Embed)

☠ b) CUBE (-> **ClU**ster and em**BE**d)

~ c) Spot

d) TrafficSpot

e) InfoShield

• English word



# 'N': How to pick good names?



- ☠ a) CAE (-> Cluster And Embed)
- ☠ b) CUBE (-> **ClU**ster and em**BE**d)
- ~ c) Spot
- d) TrafficSpot
- e) InfoShield

# 'N': How to pick good names?



- ☠ a) CAE (-> Cluster And Embed)
- ☠ b) CUBE (-> **ClU**ster and em**BE**d)
- ~ c) Spot
- ✓ d) TrafficSpot
- e) InfoShield

# 'N': How to pick good names?



- ☠ a) CAE (-> Cluster And Embed)
- ☠ b) CUBE (-> **ClU**ster and em**BE**d)
- ~ c) Spot
- ✓ d) TrafficSpot
- e) InfoShield

# 'N': How to pick good names?



- ☠ a) CAE (-> Cluster And Embed)
- ☠ b) CUBE (-> **ClU**ster and em**BE**d)
- ~ c) Spot
- ✓ d) TrafficSpot
- ✓ e) InfoShield

# Recipes for ‘good names’ – part1/4

clusterEmbed

TrafficLight

# Recipes for ‘good names’ – part1/4

1) ‘What’ – not ‘how’

clusterEmbed

TrafficLight



# Recipes for ‘good names’ – part 1/4



1) ‘What’ – not ‘how’

clusterEmbed

TrafficLight



# Recipes for ‘good names’ – part2/4

Heraclitus

TripWire

ZX23-S



# Let's take a small break

Joke

Q: What is a professor?



# Let's take a small break

Joke

Q: What is a professor?

A: The person who talks while you sleep.



# Let's take a small break

Joke

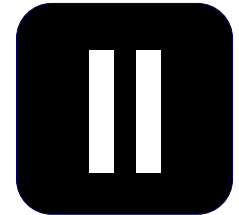
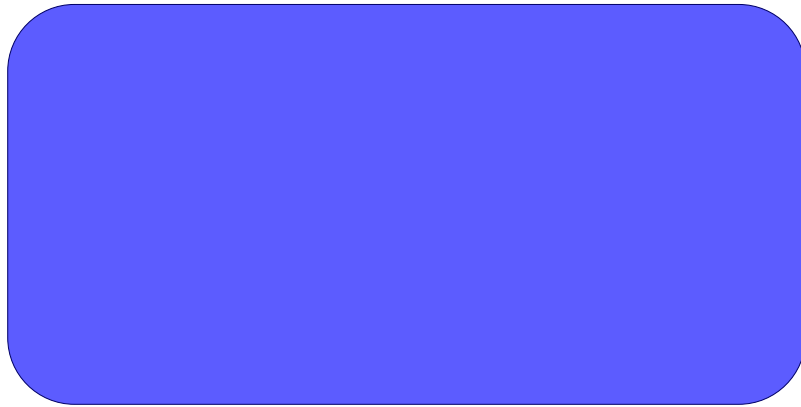
Q: What is a professor?

A: The person who talks while you sleep.



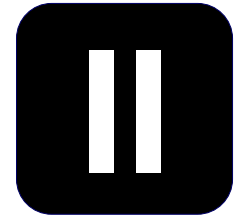
*Painfully true, in my experience  
as a professor – sigh!*

# Recipes for 'good names' – part 2/4



# Recipes for 'good names' – part 2/4

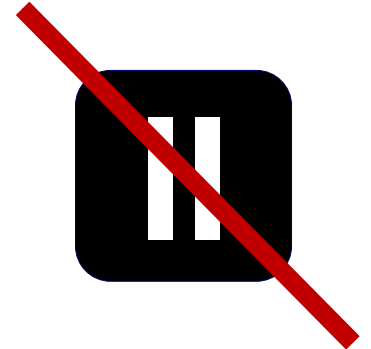
Heraclitus



# Recipes for 'good names' – part 2/4

Heraclitus

TripWire



# Recipes for ‘good names’ – part2/4

Heraclitus

TripWire

ZX23-S

# Recipes for 'good names' – part2/4

2) Easy to pronounce/remember

Heraclitus

TripWire

ZX23-S



# Recipes for 'good names' – part 2/4

2) Easy to pronounce/remember

Heraclitus

TripWire

ZX23-S



# Recipes for ‘good names’ – part3/4

SOAR

SkySoar

# Recipes for 'good names' – part3/4

3) But NOT an English word (google collisions..)

SOAR

SkySoar

# Recipes for 'good names' – part3/4

3) But NOT an English word (google collisions..)

SOAR

SkySoar

Google



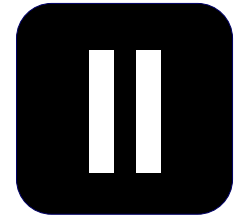
# Recipes for ‘good names’ – part4/4

EaglesEye

MAFIA

Poltergeist

Sherlock



# Recipes for 'good names' – part4/4

## 4) Positive connotation

EaglesEye

MAFIA

Poltergeist

Sherlock

# Recipes for ‘good names’ – part4/4

## 4) Positive connotation

MAFIA

EaglesEye

Poltergeist

Sherlock



# Recipes for 'good names' – part 4/4

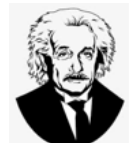
## 4) Positive connotation

MAFIA

Poltergeist

EaglesEye

Sherlock





# Recipes for ‘good names’



1) ‘What’ – not ‘how’

clusterEmbed

TrafficLight



2) Easy to pronounce/remember

Heraclitus

TripWire



ZX23

3) But NOT an English word (google collisions..)

SOAR

SkySoar

Google

4) Positive connotation

MAFIA

EaglesEye

Poltergeist

Sherlock



# Recipes for ‘good names’



1) ‘What’ – not ‘how’

clusterEmbed

TrafficLight



2) Easy to pronounce/remember

Heraclitus

TripWire



ZX23

3) But NOT an English word (google collisions..)



SOAR

SkySoar

Google

4) Positive connotation



MAFIA

EaglesEye

Poltergeist

Sherlock



## Drill: pick a name!



Suppose that you have developed a method that detects Twitter users that exhibit similar behavior. You use random seeds, and belief propagation, to spot such groups.



- *SSTU* (‘spotting similar twitter user’) )
- *BPSim* (Belief Propagation for similar user detection)
- *TwinSpot*

## Drill: pick a name!



Suppose that you have developed a method that detects Twitter users that exhibit similar behavior. You use random seeds, and belief propagation, to spot such groups.

- ✗ • *SSTU* (‘spotting similar twitter users’)
- ✗✓ • *BPSim* (Belief Propagation for similar user detection)
- ✓ • *TwinSpot*

# Drill: pick a name!

✓✓• *CopyCatch* (!!)

[\[Alex Beutel+, WWW13\]](#)



# Drill: pick a name!

✓✓• *CopyCatch* (!!)

[\[Alex Beutel+, WWW13\]](#)



👉 team up with a native speaker 😊

# Summary of summary: F.A.N.



- **F**igure (one or more; ‘what’, not ‘how’)



	Constant	Slot	Insertion	Deletion	Substitution
$T_1$ today stats	*				
#1 today stats	no new followers			one follower	no unfollowers via httpcodjgxbviq
#2 today stats	followers				no unfollowers via httpcodjgxbviq
#3 today stats				one follower	no unfollowers via httpcodjgxbviq
#4 today stats				one follower	unfollowers via httpcodjgxbviq

- **A**sk a ‘rhetorical’ question (and answer it!)



- **N**ame your method

**CopyCatch**



**ClusterEmbed**



# Outline



- Top 3 lessons: ‘F.A.N.’
  - Figure (‘crown jewel’ figure)
  - Ask (ask a rhetorical question)
  - Name (give a name to your method/system)



- 
- ➔ • A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions





# Think like a reviewer

Reviewers:

- Pressed for time
- Un-paid / un-thanked



Q: what should authors do?

# Think like a reviewer

Reviewers:

- Pressed for time
- Un-paid / un-thanked



Q: what should authors do?

- A1: **respect** & save their time  
(using figures, tables, **emphasis**)
- A2: attention routing

# Think like a reviewer

Reviewers:

- Pressed for time
- Un-paid / un-thanked



Q: what should authors do?

- A1: **respect** & save their time  
(using figures, tables, **emphasis**)
- A2: attention routing

**All F.A.N. recipes do that**

# Outline



- Top 3 lessons: ‘F.A.N.’
  - **F**igure (‘crown jewel’ figure)
  - **A**sk (ask a rhetorical question)
  - **N**ame (give a name to your method/system)



- 
- A step back: ‘think like a reviewer’
  - More battle scars; and ~10 remedies
  - Conclusions



# More recipes

- Check template tar-file with ‘orange suggestions’:  
<https://www.cs.cmu.edu/~christos/MetaPaper>
- `tar xfv; make`



*Christos' template - v14: OURMETHOD: Spotting Fake Reviews in App Stores*

first name SCS CMU first@cs.cmu.edu	second name SCS CMU second@andrew.cmu.edu	Christos Faloutsos SCS CMU christos@cs.cmu.edu
---	---	--

December 15, 2020

**Abstract**

*Historical question:* - What is the best rhetorical question you can start with?

How can you find strange nodes in a who-calls-whom graph? Spotting anomalies in graphs is an important topic.

*what - NOT 'how'* - List the benefits of the approach - NOT the details of how you do it!

**Our OURMETHOD method**

*Figure:* Give a NAME to the method - ideal name should (a) be an english-like word, but NOT a vocabulary word (b) easy to pronounce (say it three times, quickly) (c) should emphasize the main idea/insight/advantage of your method (NOT the steps you took - 'FrankSpot' is good, 'DeepLearnFraud' is not) (d) should have positive connotation ('eagle', 'lion', 'safe', 'guard', 'spot', 'alert')

has the following properties (a) Scalability, being linear on the input size (b) Effectiveness, spotting 90% of the anomalies in real data (c) *Parameter-free*, requiring no user-defined parameters.

**numbers:** Mention some performance numbers.

Experiments on 3GB of real data from opinions.com illustrate the benefits of our method.

**1 Introduction**

*Again, a rhetorical question*

Given a large count of reviews for products, how can one spot the fake ones. On line reviews are im-

portant. They are often faked, for monetary gain. How to spot the truth?

Here we propose OURMETHOD, a method to spot fake reviews. The main idea behind our method is a principled way to merge several warning signals. Figure 1 shows the results of our method

**Crown jewel:** show-case our very best results, easy to understand

**2-word-tag:** for each figure caption, give the 2-word conclusion in bold

where OURMETHOD outperforms the competition by up to 999%.

The advantages of our method are

- **Scalability** : it scales linearly with the input size
- **Effectiveness** : it gives very good reconstruction error, on real data

Figure 1: **OURMETHOD wins**: Execution time for OURMETHOD, on opinions.com

# More recipes

- Check template tar-file with ‘orange suggestions’:  
<https://www.cs.cmu.edu/~christos/MetaPaper>
- `tar xfv; make`

*rhetorical question:* - What is the best rhetorical question you can start with?

Christos' template - v14: OURMETHOD: Spotting Fake Reviews in App Stores

first name SCS CMU first@cs.cmu.edu	second name SCS CMU second@andrew.cmu.edu	Christos Faloutsos SCS CMU christos@cs.cmu.edu
---	---	--

December 15, 2020

**Abstract**

*rhetorical question:* - What is the best rhetorical question you can start with?

How can you find strange nodes in a who-calls-whom graph? Spotting anomalies in graphs is an important topic.

*what - NOT 'how'*: List the benefits of the approach - NOT the details of how you do it!

Our OURMETHOD method

*Repeat:* Give a NAME to the method - ideal name should (a) be an english-like word, but NOT a vocabulary word (b) easy to pronounce (say it three times, quickly) (c) should emphasize the main idea/insight/advantage of your method (NOT the steps you took - 'FrankSpoc' is good, 'DeepLearnFraud' is not) (d) should have positive connotation ('eagle', 'lion', 'safe', 'guard', 'spic', 'alert')

has the following properties (a) Scalability, being linear on the input size (b) Effectiveness, spotting 90% of the anomalies in real data (c) *Parameter-free*, requiring no user-defined parameters.

*numbers:* Mention some performance numbers.

Experiments on 3GB of real data from opinions.com illustrate the benefits of our method.

**1 Introduction**

*Again, a rhetorical question*

Given a large count of reviews for products, how can one spot the fake ones. On line reviews are im-

portant. They are often faked, for monetary gain. How to spot the truth?

Here we propose OURMETHOD, a method to spot fake reviews. The main idea behind our method is a principled way to merge several warning signals. Figure 1 shows the results of our method

*Crown jewel:* show-case our very best results, easy to understand

*2-word-tag:* for each figure caption, give the 2-word conclusion in bold

where OURMETHOD outperforms the competition by up to 99%.

The advantages of our method are

- **Scalability** : it scales linearly with the input size
- **Effectiveness** : it gives very good reconstruction error, on real data

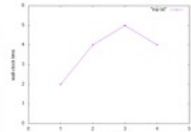


Figure 1: OURMETHOD wins. Execution time for OURMETHOD, on opinions.com

# More recipes – part 1 of 4

## 1. Title:

- ✓ a. Name your method (‘what’, not ‘how’)

## 2. Abstract

- ✓ a. Ask a rhetorical question
- b. Give performance numbers



# More recipes - part 2 of 4



## 3. Intro:

- ✓ a. Again, rhetorical question
- ✓ b. ‘crown jewel’ **f**igure
- c. List (bullets) 2-4 contributions
- d. (Informal) problem definition
- e. Give two-word summary for each contribution

## 4. Literature survey

- a. ‘salesman matrix’: rows are features; columns are baselines



# More recipes - part 3 of 4

## 5. Proposed method

- a. **No citations** from now on
- b. Clear problem definition
- c. Add theorems/lemmas and proofs ('QED')

## 6. Experiments

- a. Each sub-section should confirm each of the contributions



# More recipes - part 3 of 4

## 7. Conclusions

- a. Repeat the contributions from the intro

## 8. Globally:

- a. ‘two-word tag’, for every figure/table caption

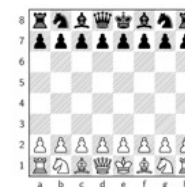
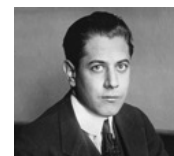


# F.A.Q.

- Are these guidelines mandatory/necessary?
- Are they enough/sufficient?

# F.A.Q.

- Are these guidelines mandatory/necessary?
  - **NO** (but help: + epsilon)
- Are they enough/sufficient?
  - **NO** (more, every year)



# Resources



1. Human-trafficking detection paper:

<http://catvajiac.me/files/infoshield.pdf>

*INFOSHIELD: Generalizable Information-Theoretic*

*Human-Trafficking Detection*, Meng-Chieh Lee, Catalina Vajiac, et al, ICDE 2021, Chania, Greece



2. Check ‘orange suggestions’ at:

<https://www.cs.cmu.edu/~christos/MetaPaper/>



# Conclusions – high level

1) **Respect reviewers’ time**



2) ‘what’, not ‘how’



# Conclusions - detailed



- F.A.N. (Figure; Ask question; Name your method)

	Constant	Slot	Insertion	Deletion	Substitution
T <sub>1</sub> today stats	*				
#1 today stats	no new followers			unfollowers via httpcodjgxbviq	
#2 today stats	followers			no unfollowers via httpcodjgxbviq	
#3 today stats		one follower		no unfollowers via httpcodjgxbviq	
#4 today stats		one follower		unfollowers via httpcodjgxbviq	



- (check orange suggestions)



# Conclusions - detailed



- F.A.N. (**F**igure; **A**sk question; **N**ame your method)

	Constant	Slot	Insertion	Deletion	Substitution
T <sub>1</sub> today stats	*				
#1 today stats	no new followers			one follower	no unfollowers via httpcodjgxbviq
#2 today stats	followers				no unfollowers via httpcodjgxbviq
#3 today stats				one follower	no unfollowers via httpcodjgxbviq
#4 today stats				one follower	unfollowers via httpcodjgxbviq



‘ACK: ... we used the MetaPaper ...’

[christos@cs.cmu.edu](mailto:christos@cs.cmu.edu)

<https://www.cs.cmu.edu/~christos/MetaPaper/>



# Conclusions - detailed



• F.A.N. (Figure; Ask question; Name your method)

	Constant	Slot	Insertion	Deletion	Substitution
T <sub>1</sub> today stats	*	one follower	no unfollowers	via httpcodjgxbviq	
#1 today stats	no new followers		unfollowers	via httpcodjgxbviq	
#2 today stats	followers		no unfollowers	via httpcodjgxbviq	
#3 today stats		one follower	no unfollowers	via httpcodjgxbviq	
#4 today stats		one follower	unfollowers	via httpcodjgxbviq	

