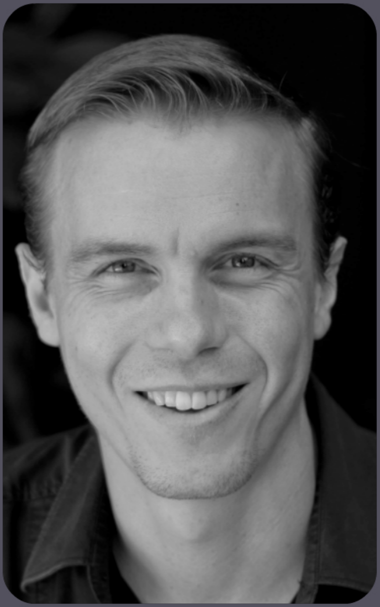
hacker bits

August 2016



new bits

Hello from Redmond!

As we near the dog days of summer, we can't think of anything more restorative than catching up on our reading with a tall glass of iced tea. And if you are looking for something worthwhile to read, don't miss Cody Littlewood's *My first 10 minutes on a server*, an essential primer for anyone looking to secure Ubuntu.

Speaking of summer reading, some of you have written to tell us that you'd love to see more data-related articles…well, consider that done! We'll be bringing you lots of informative data-related pieces in the future, so stay tuned!

Before we sign off (and get back to our iced tea), we'd like to give a big shout-out to our new subscribers (we're looking at you, fans of DevMastery)! Welcome and we hope you like what you see at *Hacker Bits*. Our mission at *Hacker Bits* is to help our readers *learn more*, *read less* and *stay current*, so feel free to let us know how we can do it better!

Peace and stay cool!

— Maureen and Ray

us@hackerbits.com

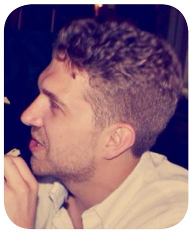
content bits

August 2016

contributor bits

Matthew Green

Matthew is a cryptographer and professor at Johns Hopkins University. He has designed and analyzed cryptographic systems used in wireless networks, payment systems and digital content protection platforms.



Cody Littlewood

Cody is CEO at Codelitt Incubator, a corporate skunkworks, R&D program and product incubator. He is a staunch supporter of the EFF, Mozilla, The Linux Foundation, and other open source, net freedom and privacy focused organizations.

Troy Hunt

Troy is a Pluralsight author, Microsoft Regional Director, Most Valuable Professional (MVP) and world-renowned Internet security specialist. He’s the creator of “Have I been pwned?”, the free online service for breach monitoring and notifications, and blogs at troyhunt.com from his home in Australia.

Gabriel Weinberg

Gabriel Weinberg is the CEO & Founder of DuckDuckGo, the search engine that doesn't track you. I also co-authored Traction, the book that helps you get traction. He resides in Valley Forge, PA and on Twitter @yegg.

Mark Adler

Dr. Mark Adler is a Fellow at NASA's Jet Propulsion Laboratory, where he has been to Saturn and Mars virtually through robot spacecraft. He has been a key contributor to the opensource Info-ZIP, gzip, and zlib projects.

Per Harald Borgen

Per is a developer and machine learning enthusiast from Norway. He currently works as a developer at Xeneta. He previously co-founded a kids app startup, Propell, which he ran for 3 years as the CEO. He has a degree in macro economics and learned to code at Founders&Coders in London in 2015.

Aline Lerner

Aline is the co-founder and CEO of interviewing.io. She likes ranting on the Internet about how hiring is broken, and her work has appeared in Forbes, the Wall Street Journal, and Fast Company.

Shyal Beardsley

Shyal started his career doing R&D in Visual FX, including working on the Harry Potter movies. He is now CTO for a real estate startup. He is also the owner at shyal.com ltd. where he focuses on rapid development and entrepreneurship.

Ray Li
Curator

Ray is a software engineer and data enthusiast who has been blogging at rayli.net for over a decade. He loves to learn, teach and grow. You'll usually find him wrangling data, programming and lifehacking.

Maureen Ker
Editor

Maureen is an editor, writer, enthusiastic cook and prolific collector of useless kitchen gadgets. She is the author of 3 books and 100+ articles. Her work has appeared in the *New York Daily News*, and various adult and children's publications.

What is differential privacy?

By MATTHEW GREEN

Apple announced that they will be using a technique called "Differential Privacy" to improve the privacy of their data collection practices.

At the 2016 WWDC keynote, Apple announced a series of new security and privacy features, including one feature that's drawn a bit of attention and confusion. Specifically, Apple announced that they will be using a technique called "Differential Privacy" (henceforth referred to as DP) to improve the privacy of their data collection practices.

The reaction to this by most people has been a big "???", since few people have even heard of Differential Privacy, let alone understand what it means. Unfortunately Apple isn't known for being terribly open when it comes to sharing the secret sauce that drives their platform, so we'll just have to hope that at some point they'd decide to publish more.

What we know so far comes from Apple's iOS 10 Preview guide:

> *Starting with iOS 10, Apple is using Differential Privacy technology to help discover the usage patterns of a large number of users without compromising individual privacy.*
>
> *To obscure an individual's identity, Differential Privacy adds mathematical noise to a small sample of the individual's usage pattern. As more people share the same pattern, general patterns begin to emerge, which can inform and enhance the user experience.*
>
> *In iOS 10, this technology will help improve QuickType and emoji suggestions, Spotlight deep link suggestions and Lookup Hints in Notes.*

To make a long story short, it sounds like Apple is going to be collecting a lot more data from your phone. They're mainly doing this to make their services better, and not to collect individual users' usage habits.

To guarantee this, Apple intends to apply sophisticated statistical techniques to ensure that this aggregate data the statistical functions it computes over all your information — don't leak your individual contributions. In principle this sounds pretty good. But of course, the devil is always in the details.

While we don't have those details, this seems like a good time to at least talk a bit about what Differential Privacy is, how it can be achieved, and what it could mean for Apple and for your iPhone.

The motivation

In the past several years, "average people" have gotten used to the idea that they're sending a hell of a lot of personal information to the various services they use. Surveys also tell us they're starting to feel uncomfortable about it.

This discomfort makes sense when you think about companies using our personal data to market to us.

But sometimes there are decent motivations for collecting usage information. For example, Microsoft recently announced a tool that can diagnose pancreatic cancer by monitoring your Bing queries. Google famously runs Google Flu Trends. And of course, we all benefit from crowdsourced data that improves the quality of the services we use — from mapping applications to restaurant reviews.

Unfortunately, even well-meaning data collection can go bad. For example, in the late 2000s, Netflix ran a competition to develop a better film recommendation algorithm. To drive the competition, they released

> ***The neat thing about DP is (that it) can be applied to...complex statistical calculations like the ones used by Machine Learning algorithms.***

an "anonymized" viewing dataset that had been stripped of identifying information.

Unfortunately, this de-identification turned out to be insufficient. In a well-known piece of work, Narayanan and Shmatikov showed that such datasets could be used to re-identify specific users and even predict their political affiliation (!), if you simply knew a little bit of additional information about a given user.

This sort of thing should be worrying to us. Not just because companies routinely share data (though they do) but because breaches happen, and even statistics about a dataset can sometimes leak information about the individual records used to compute it. Differential Privacy is a set of tools that was designed to address this problem.

What is Differential Privacy?

Differential Privacy is a privacy definition that was originally developed by Dwork, Nissim, McSherry and Smith, with major contributions by many others over the years. Roughly speaking, what it states can be summed up intuitively as follows:

> *Imagine you have two otherwise identical databases, one with your information in it, and one without it. Differential Privacy ensures that the probability that a statistical query will produce a given result is (nearly) the same whether it's conducted on the first or second database.*

One way to look at this is that DP provides a way to know if your data has a significant effect on the outcome of a query. If it doesn't, then you might as well contribute to the database, since there's almost no harm that can come of it.

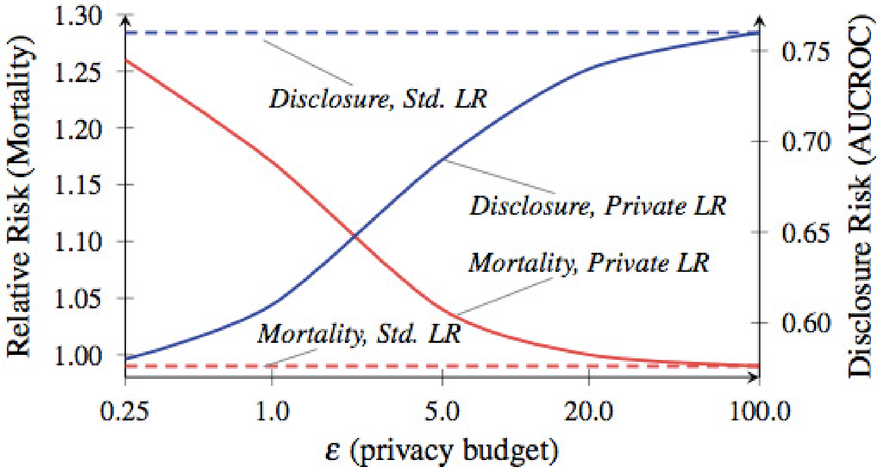
Consider a silly example: Imagine that you choose to enable a reporting feature on your iPhone that tells Apple if you like to use the ice cream emoji routinely in your iMessage conversations. This report consists of a single bit of information: 1 indicates you like ice cream, and 0 doesn't. Apple might receive these reports and fill them into a huge database. At the end of the day, it wants to be able to derive a count of the users who like this particular emoji.

It goes without saying that the simple process of "tallying up the results" and releasing them does not satisfy the DP definition, since computing a sum on the database that contains your information will potentially produce a different result from computing the sum on a database without it.

Thus, even though these sums may not seem to leak *much* information, they reveal at least a little bit about you. A key observation of the Differential Privacy research is that in many cases, DP can be achieved if the tallying party is willing to add random noise to the result.

For example, rather than simply reporting the sum, the tallying party can inject noise from a Laplace or Gaussian distribution, producing a result that's not quite exact, but that masks the contents of any given row. (For other interesting functions, there are many other techniques as well.)

Even more usefully, the calculation of "how much" noise to inject *can be made without knowing the contents of the database itself (or even its size).* That is, the noise calculation can be performed based only on knowledge of the function to be computed, and the acceptable amount of data leakage.



▲ Mortality vs. info disclosure, from Frederikson et al. The red line is patient mortality.

A tradeoff between privacy and accuracy

Now obviously calculating the total number of ice cream-loving users on a system is a pretty silly example. The neat thing about DP is that the same overall approach can be applied to much more interesting functions, including complex statistical calculations like the ones used by Machine Learning algorithms. It can even be applied when many different functions are all computed over the same database.

But there's a big caveat here. Namely, while the amount of "information leakage" from a single query can be bounded by a small value, this value is not zero. Each time you query the database on some function, the total "leakage" increases and can never go down. Over time, as you make more queries, this leakage can start to add up.

This is one of the more challenging aspects of DP. It manifests in two basic ways:

1. *The more information you intend to "ask" of your database, the more noise has to be injected in order to minimize the privacy leakage.* This means that in DP there is generally a fundamental tradeoff between accuracy and privacy, which can be a big problem when training complex ML models.
2. *Once data has been leaked, it's gone.* Once you've leaked as much data as your calculations tell you is safe, you can't keep going, at least not without risking your users' privacy. At this point, the best solution may be to just destroy the database and start over,if such a thing is possible.

The total allowed leakage is often referred to as a "privacy budget", and it determines how many queries will be allowed (and how accurate the results will be). The basic lesson of DP is that *the devil is in the budget.* Set it too high, and you leak your sensitive data. Set it too low, and the answers you get might not be particularly useful.

Now in some applications, like many of the ones on our iPhones, the lack of accuracy isn't a big deal. We're used to our phones making mistakes. But sometimes when DP is applied in complex applications, such as training Machine Learning models, this really does matter.

To give an *absolutely crazy example* of how big the tradeoffs can be, consider this paper by Frederikson et al. from 2014. The authors began with a public database linking warfarin dosage outcomes to specific genetic markers. They then used ML techniques to develop a dosing model based on their database, but applied DP at various privacy budgets while training the model. Then they evaluated both the information leakage and the model's success at treating simulated "patients."

The results showed that the model's accuracy depends a lot on the privacy budget on which

Randomized response has been shown to achieve Differential Privacy, with specific guarantees that can be adjusted by fiddling with the coin bias.

it was trained. If the budget is set too high, the database leaks a great deal of sensitive patient information, but the resulting model makes dosing decisions that are about as safe as standard clinical practice.

On the other hand, when the budget was reduced to a level that achieved meaningful privacy, the "noise-ridden" model had a tendency to kill its "patients."

Now before you freak out, let me be clear: *your iPhone is not going to kill you*. Nobody is saying that this example even vaguely resembles what Apple is going to do on the phone.

The lesson of this research is simply that there are interesting tradeoffs between effectiveness and the privacy protection given by any DP-based system. These tradeoffs depend to a great degree on *specific* decisions made by the system designers, the parameters chosen by the deploying parties, and so on. Hopefully Apple will soon tell us what those choices are.

How do you collect the data, anyway?

You'll notice that in each of the examples above, I've assumed that queries are executed by a trusted database operator who has access to all of the "raw" underlying data. I chose this model because it's the traditional model used in most of the literature, not because it's a particularly great idea.

In fact, it would be worrisome if Apple was actually *implementing* their system this way. That would require Apple to collect all of your raw usage information into a massive centralized database, and then ("trust us!") calculate privacy-preserving statistics on it. At a minimum this would make your data vulnerable to subpoenas, Russian hackers, nosy Apple executives and so on.

Fortunately this is not the only way to implement a Differentially Private system. On the theoretical side, statistics can be computed using fancy cryptographic techniques (such as secure multi-party computation or fully-homomorphic encryption.) Unfortunately these techniques are probably too inefficient to operate at the kind of scale Apple needs.

A much more promising approach is not to collect the raw data at all. This approach was recently pioneered by Google to collect usage statistics in their Chrome browser. The system, called RAPPOR, is based on an implementation of the 50-year old randomized response technique. Randomized response works as follows:

1. When a user wants to report a piece of potentially embarrassing information (made up example: "Do you use Bing?"), they first flip a coin, and if the coin comes up "heads", they return a random answer, calculated by flipping a second coin. Otherwise they answer honestly.
2. The server then collects answers from the entire population, and (knowing the probability that the coins will come up "heads"), adjusts for the included "noise" to compute an approximate answer for the true response rate.

Intuitively, randomized response protects the privacy of individual user responses, because a "yes" result could mean that you use Bing, or it could just be the effect of the first mechanism (the random coin flip).

More formally, randomized response has been shown to achieve Differential Privacy, with

▲ I've met Craig Federighi. He actually looks like this in person.

specific guarantees that can be adjusted by fiddling with the coin bias.

RAPPOR takes this relatively old technique and turns it into something much more powerful. Instead of simply responding to a single question, it can report on complex vectors of questions, and may even return complicated answers, such as strings, fore.g., which default homepage you use.

The latter is accomplished by first encoding the string into a Bloom filter — a bitstring constructed using hash functions in a very specific way. The resulting bits are then injected with noise, and summed, and the answers recovered using a (fairly complex) decoding process.

While there's no hard evidence that Apple is using a system like RAPPOR, there are some small hints. For example, Apple's Craig Federighi describes Differential Privacy as "*using hashing, subsampling and noise injection to enable… crowdsourced learning while keeping the data of individual users completely private.*" That's pretty weak evidence for anything, admittedly, but the presence of "hashing" in that quote at least hints towards the use of RAPPOR-like filters.

The main challenge with randomized response systems is that they can leak data if a user answers the same question multiple times. RAPPOR tries to deal with this in a variety of ways, one of which is to identify static information and thus calculate "permanent answers" rather than re-randomizing each time.

But it's possible to imagine situations where such protections could go wrong. Once again, the devil is very much in the details — we'll just have to see. I'm sure many fun papers will be written either way.

So is Apple's use of DP a good thing or a bad thing?

As an academic researcher and a security professional, I have mixed feelings about Apple's announcement. On the one hand, as a researcher, I understand how exciting it is to see research technology actually deployed in the field. And Apple has a very big field.

On the flipside, as security professionals, it's our job to be skeptical, and at a minimum demand people release their security-critical code (as Google did with RAPPOR), or at least to be straightforward about what it is they're deploying.

If Apple is going to collect significant amounts of new data from the devices that we depend on so much, we should really make sure they're doing it right, rather than cheering them for Using Such Cool Ideas. (I made this mistake already once, and I still feel dumb about it.)

But maybe this is all too "inside baseball". At the end of the day, it sure looks like Apple is *honestly trying to do something to improve user privacy*, and given the alternatives, maybe that's more important than anything else. ■

Reprinted with permission of the original author. First appeared at blog.cryptographyengineering.com.

My first 10 minutes on a server — primer for securing Ubuntu

By CODY LITTLEWOOD

I check our logwatch email every morning and thoroughly enjoy watching several hundred of attempts at gaining access with little prevail.

“My First 5 Minutes on a Server,” by Bryan Kennedy, is an excellent intro into securing a server against most attacks. We have a few modifications to his approach that we wanted to document as part of our efforts of externalizing our processes and best practices. We also wanted to spend a bit more time explaining a few things that younger engineers may benefit from.

I check our logwatch email every morning and thoroughly enjoy watching several hundred (sometimes 1000s) of attempts at gaining access with little prevail. (Many are rather unimaginative, such as trying `root` with password `1234` over and over again).

This general overview works for Debian/Ubuntu servers, which are our personal favourite choice. These usually only serve as hosts for Docker containers, but the principles still apply. We'll go more in depth in locking down a server specifically for use as a Docker host another time.

On large scale, you'll be better off with a full automated setup using something like Ansible or Shipyard. However, sometimes you're just creating a single server or working on a base for an Ansible recipe, which is what this is meant to cover.

Disclaimer: This is meant to serve as a primer and a base. You should extend upon it as your needs dictate.

First things first

We don't even have a password for our root user yet. We'll want to select something random and complex. We use a password manager's password generator set to the most difficult setting. The PW manager saves the password and it is encrypted with access only given by a long master password.

A couple of redundancies are provided here (long, complex, random password + password is stored behind encryption/another long password). Whether you use a PW manager or some other means, keep this safe and behind some form of encryption. You'll only need this root password if you lose your sudo password.

```
# passwd
```

Note: There's a lot of discussion going on both HN and Reddit on root passwords. It's worth a read.

Next, you'll need to update the repositories and upgrade your system applying the latest patches. We'll have a section for how to automate security upgrades later on.

```
apt-get update
apt-get upgrade
```

Add your user

You should never be logging on to a server as root. We follow a similar convention as Bryan in our user name, but you could use whatever convention you'd like. With a small team, having one login user hasn't been an issue for us, but with a larger team, best practice would dictate that different users would be set up with different levels of permission only granting sudo permissions to a select few.

```
useradd deploy
mkdir /home/deploy
mkdir /home/deploy/.ssh
chmod 700 /home/deploy/.ssh
```

Set up your preferred shell for the deploy user. Here we use bash:

```
usermod -s /bin/bash deploy
```

Remember `chmod 700` means that owner can read, write, and execute. We're still root but in a minute we'll recursively `chown`

> ***SSH keys are better than passwords only because they contain and require more information.***

this folder for the deploy user and deploy group. Only this user should have access to do anything with the .ssh folder.

Require ssh key authentication

We tend to avoid passwords for logging into servers. There was a lot of discussion around this after Bryan's original guide came out, but I tend to fall into this camp as well. Here are a few notes on this:

1. SSH keys are better than passwords only because they contain and require more information.
2. Passwords can be brute forced. Guessing a public key is so essentially impossible that they can be considered perfectly secure.
3. What about a stolen machine? Yes, they have your private key, but expiring an SSH key is easy – just remove the public key from authorized_keys. You should also have your private key protected by a secure and long passphrase. See next point.
4. All of this works, AS LONG AS YOU HAVE A LONG AND SECURE PASSPHRASE PROTECTING YOUR KEY. Repeated because it's bloody important.

So let's make password authentication a thing of the past on our server. Copy the contents of your `id_rsa.pub`[1] on your local machine to your server's authorized keys file.

```
vim /home/deploy/.ssh/authorized_keys
```

Let's set the right permissions based on the Linux security principal of least privilege:

```
chmod 400 /home/deploy/.ssh/authorized_keys

chown deploy:deploy /home/deploy -R
```

`chmod 400` sets permissions so that the file can be read by owner. The second command, `chown` makes the user deploy and group deploy owners (recursively) of their home directory. We referenced this earlier when setting read/write/execute permissions to owner for this directory.

We're going to come back in a second after we've properly tested our deploy user and sudo to disable logging in as the root user, and enforce ssh key logins only.

Test deploy user and set up sudo

We're going to test logging in as deploy, while keeping our ssh connection as `root` open just in case. If it works, we'll use our open connection as `root` user to set a password for deploy. Since we're disabling password logins, this password will be used when sudo-ing.

Again we use a pw manager to create a complex and random password, saving it behind an encrypted wall, and sharing it among the team (syncing the encrypted pw file).

```
passwd deploy
```

Setting up sudo is simple. Open up the sudo file with:

```
visudo
```

Add the `%sudo` group below the `root` user as shown below. Make sure to comment out any other users and groups with a `#`. (users have no prefix and groups start with %.) Most fresh installs won't have any there, but just make sure.

```
root     ALL=(ALL) ALL
%sudo    ALL=(ALL:ALL) ALL
```

Then add `deploy` user to the `sudo` group.

Simple is generally better with security. The DigitalOcean ufw is really good and goes over the basics.

```
usermod -aG sudo deploy
```

`deploy` now has access to sudo permissions. Now normally you need exit and re-login to the shell in order to start having access to the group's permissions. There's a little trick though to avoid having to do that:

```
exec su -l deploy
```

This starts a new interactive shell for the `deploy` user with the new permissions to the `sudo` group. It will require your `deploy`'s password, but it feels faster than logging out and logging back in.

Note: Thanks to ackackacksyn on Reddit for pointing out that you should not add users directly to sudoers. Thanks to [FredFS456](#) in /r/netsec for pointing out you need to logout and log back in for group permissions to take effect.

Enforce SSH key logins

SSH configuration for the machine is stored here:

```
vim /etc/ssh/sshd_config
```

You'll want to change these lines (or add if missing) in the file to match below. I think they're pretty self-explanatory. You'll want to add the IP that you use to connect. We have a company VPN setup with OpenVPN with cryptographic authentication, so in order to connect to a server, you must also be authenticated and connected to the VPN.

```
PermitRootLogin no

PasswordAuthentication no

AllowUsers deploy@(your-VPN-
or-static-IP)

AddressFamily inet
```

Enable all these rules by restarting the ssh service. You'll probably need to reconnect (do so by using your deploy user!)

```
service ssh restart
```

Note: Thanks to raimue and mwpmaybe on HN for pointing out that [fail2ban (installed later) does not support IPv6 right now](#), so I added `AddressFamily inet` to the `sshd_config` file, which will only allow IPv4 (which fail2ban does support).

Setting up a firewall

There are usually two camps. Those who use IPtables directly and those who use a handy interface called ufw, which is a layer on top of IPtables meant to simplify things. Simple is generally better with security. The DigitalOcean `ufw` is really good and goes over the basics.

`ufw` is installed by default on Ubuntu and is just an `apt-get install ufw` away on Debian.

By default `ufw` should deny all incoming connections and allow all outgoing connections, however, it won't be running (because otherwise how would you be connected?). We'll go through and explicitly allow the connections that we deem okay.

First we'll want to make sure that we are supporting IPv6. Just open up the config file.

```
vim /etc/default/ufw
```

Set IPv6 to yes.

```
IPV6=yes
```

For the rest of the ports that we're going to open up, we can just use the ufw tool from command line, which is very handy.

```
sudo ufw allow from {your-ip}
to any port 22

sudo ufw allow 80
sudo ufw allow 443
sudo ufw disable
sudo ufw enable
```

The first one is a redundancy

Fail2ban is a great package that actively blocks suspicious activity as it occurs.

measure that makes sure that only connections from our IP can connect via SSH (standard SSH port).[2] While the second and third open up http and https traffic.

Note: Thanks to chrisfosterelli for pointing out that if you're going to set up the first rule (and you should), make sure you have a static IP or secure VPN that you connect to. A dynamic IP will leave you locked out of your box some day in the future.

Automated security updates

I like these. They're not perfect, but it's better than missing patches as they come out.

```
apt-get install unattended-upgrades

vim /etc/apt/apt.conf.d/10periodic
```

Update this file to match this:

```
APT::Periodic::Update-Package-Lists "1";

APT::Periodic::Download-Upgradeable-Packages "1";

APT::Periodic::AutocleanInterval "7";

APT::Periodic::Unattended-Upgrade "1";
```

I generally agree with Bryan that you'll want to disable normal updates and only enable security updates. The idea here is that you don't want an application going down without you knowing about it because some package was updated that it relies on, while security patches very rarely create dependency nightmares at an application level.

```
vim /etc/apt/apt.conf.d/50unattended-upgrades
```

Make the file look like this:

```
Unattended-Upgrade::Allowed-Origins {
    "Ubuntu lucid-security";
    //"Ubuntu lucid-updates";
};
```

You're all set.

Fail2ban

Fail2ban is a great package that actively blocks suspicious activity as it occurs. From their wiki Fail2ban scans log files (e.g. `/var/log/apache/error_log`) and bans IPs that show the malicious signs – too many password failures, seeking for exploits, and etc. It does this by adding rules to `iptables`.

Out of the box Fail2Ban comes with filters for various protocols (HTTPS, SMTP, SSH, etc). It also has integration with a lot of services like Apache and Nginx, which can provide a certain level of DDoS or brute force attack protection.

You'll want to be careful with using it in this way though because depending on the IP address where the DDoS is coming from, you could lock out real users for a time as well. It offers a lot of configuration options including integration with SendMail to notify you when an IP gets banned. Feel free to take a look at some of the links and see if any of the other options interest you.

We're just going to install it and leave the default settings for SSH as a base starting point though:

```
apt-get install fail2ban
```

2 factor authentication

2FA is not optional for us when building anything that has very sensitive requirements. Theoretically, if you're enforcing 2FA (on top of all these other measures), then in order to gain access to

Logwatch monitors your logfiles and when configured, sends you a daily email with the information parsed very nicely.

your server (baring application vulnerabilities), the attacker would have to have:

1. Access to your certificate and key to access VPN
2. Access to your computer to have your private key
3. Access to your passphrase for your private key
4. Access to your phone for 2FA

These are quite a few hurdles to jump. Even then to gain root access via sudo, they'd have to have deploy's password that is stored behind AES encryption (5).

Install this package:

```
apt-get install libpam-google-authenticator
```

Set up by running this command and following the instructions:

```
su deploy
google-authenticator
```

2FA is very easy and adds a great layer of security.

Logwatch

This is really more of a simple pleasure and monitoring tool that helps you see what's going on after the fact. Logwatch monitors your logfiles and when configured, sends you a daily email with the information parsed very nicely.

The output is quite entertaining to watch and you'll be surprised at how many attempts are made every day to gain access to your server. I install it if for no other reason than to show the team how important good security is.

There's a great write-up by DigitalOcean on Logwatch install and config, but if we're keeping to 10 minutes, we'll just install it and run a cron job to run it and email us daily.

```
apt-get install logwatch
```

Add a cron job:

```
vim /etc/cron.daily/00logwatch
```

Add this line to the cron file:

```
/usr/sbin/logwatch --output
 mail --mailto you@example.com
--detail high
```

All done

There you are. Your main concern and point of vulnerability after completing this will be your application and services. These are another animal entirely though.

We're making a big push to externalize our processes and best practices. If you're interested in learning more, take a look at our repository. We open source all of our policies and best practices, as well as continue to add to them there.

Have suggestions or questions? Comment below or submit a PR/issue on the Github repo! There are also a lot of really good bits of info on the Hacker News thread and /r/netsec. ■

[1] Make sure it's `.pub`. This seems to be very simple, but I've seen two people (both *not* members of our organization — it would be a quick way to stop being part of our org.) in my career, send me their private key (`id_rsa` without the .pub extension) when asking for their public keys.

[2] There's a couple of camps on whether to keep your SSH connection on a standard or non-standard port. See [here] and [here] for opposing sides.

Reprinted with permission of the original author. First appeared at codelitt.com/blog.

The “cobra effect” that is disabling paste on password fields

By TROY HUNT

Back in the day when the British had a penchant for conquering the world, they ran into a little problem on the subcontinent: cobras.

Turns out there were a hell of a lot of the buggers wandering around India and it also turned out that they were rather venomous, which didn't sit well with the colonials.

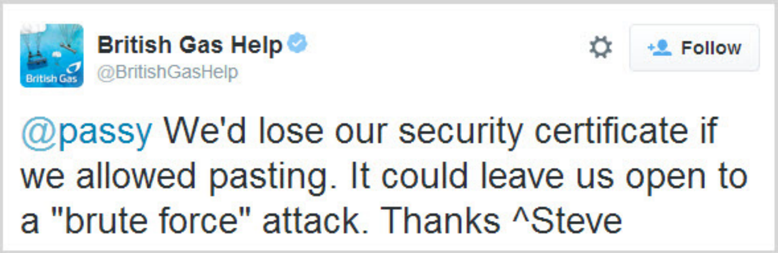
Ingenious as the British were, they decided to offer the citizens a bounty — you hand in dead cobras that would otherwise have bitten some poor imperialist and you get some cash. Problem solved.

Unfortunately, the enterprising locals saw things differently and interpreted the "cash for cobras" scheme as a damn good reason to start breeding serpents and raking in the dollars.

Having now seen the flaw in their original logical, the poms quickly scrapped the scheme, meaning no more snake bounty. Naturally the only thing for the locals to do with their now worthless cobras was to set them free so that they may seek out a nice cosy British settlement somewhere.

This became known as the cobra effect or in other words, a solution to a problem that actually makes the whole thing a lot worse.

Here's a modern day implementation of the cobra effect as it relates to the ability to paste your password into a login field:

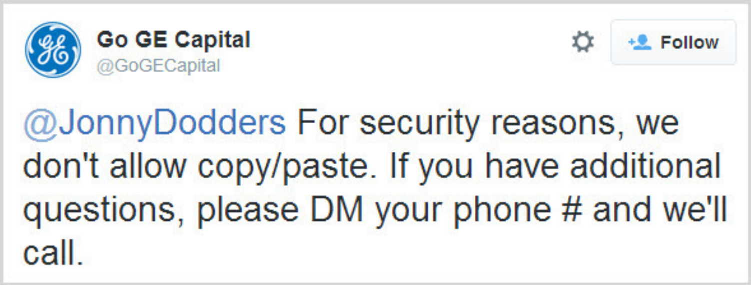


British Gas Help
@BritishGasHelp

@passy We'd lose our security certificate if we allowed pasting. It could leave us open to a "brute force" attack. Thanks ^Steve

Let's just allow the nuances of that one to sink in for a moment…

I imagine Steve was picturing armies of elite hackers working through password dictionaries with the old CTRL-C / CTRL-V and the ICO then sweeping in and taking British Gas' proudly mounted security certificate down from the wall in the foyer. That's about the best I can come up with, but let's not single out British Gas here; this is a far broader issue than just them.



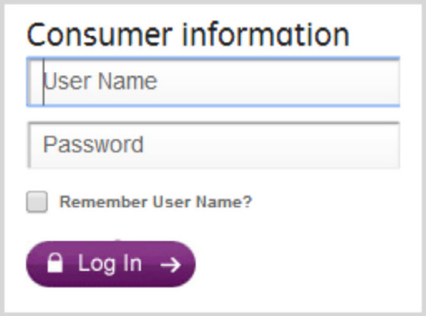
Go GE Capital
@GoGECapital

@JonnyDodders For security reasons, we don't allow copy/paste. If you have additional questions, please DM your phone # and we'll call.

For your security, we have disabled pasting of passwords

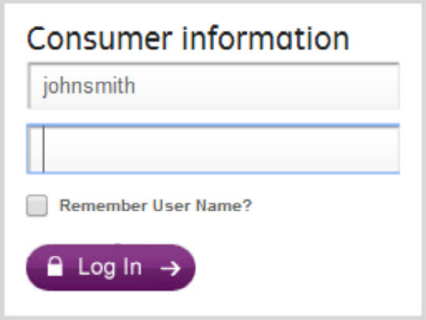
As far as amusing infosec tweets go, British Gas did a fine job but there are plenty of others who also reject the ability to paste into the password field as well, albeit with less humour (see tweet above).

But what does this actually look like? I mean a website can't disable the fact that your operating system and browser have some form of equivalent of CTRL-V, so what happens when you paste? Here's Go GE Capital before logging in:



Consumer information

Now here they are after entering the username and pasting the password:



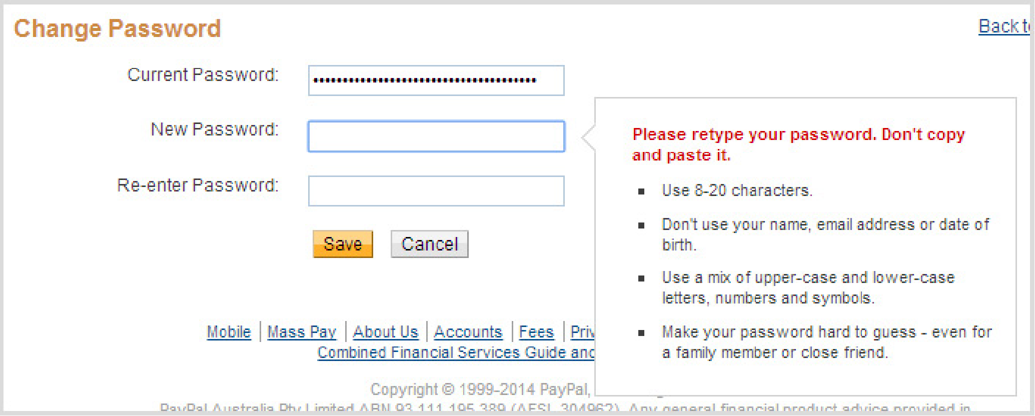
Consumer information

johnsmith

Right…

Go GE Capital just kills whatever is pasted, it disappears in the blink of an eye. No warning, no "For your safety…" message, just goneski.

PayPal take a slightly different approach on their *change password* page:



Change Password

Back t

Current Password: ••

New Password:

Re-enter Password:

Save Cancel

Please retype your password. Don't copy and paste it.

- Use 8-20 characters.
- Don't use your name, email address or date of birth.
- Use a mix of upper-case and lower-case letters, numbers and symbols.
- Make your password hard to guess - even for a family member or close friend.

Mobile | Mass Pay | About Us | Accounts | Fees | Priv

Combined Financial Services Guide and

Copyright © 1999-2014 PayPal,

PayPal Australia Pty Limited ABN 93 111 195 389 (AFSL 304962). Any general financial product advice provided in

▲ Figure 1: PayPal's change password page

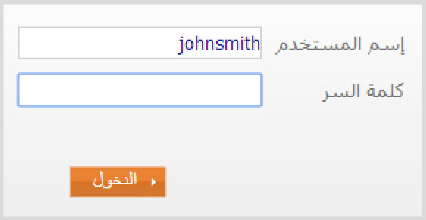
Ok, you get a message which is nice, but zero info on *why* you shouldn't be pasting in your password. I'll come back to this example later on though because there are other things going on here which help explain their rationale.

Before we go on though, let's take a quick look at the mechanics behind the anti-paste mechanism so we can better understand what goes into saving us from ourselves.

The mechanics of an anti-password-paster

Right, let's drill into this and see what's going on behind the scenes. We'll pick someone different this time. Let's try the Al Rajhi Bank in Saudi Arabia just for a change of flavour.

As you can see below, I've typed in a username and pasted a password:



إسم المستخدم johnsmith

كلمة السر

Yeah, about that password… anyway, the question remains, what voodoo is breaking this most fundamental of behaviours? For Al Rajhi, it's very simple:

```
<input type="password"
name="j_password"
class="DataPasive"
maxlength="14" size="23"
onkeypress="return
logonWithEnter(event)"
autocomplete="off"
onpaste="return false">
```

That's it — the onpaste event is returning false. Sound unfamiliar? I mean the onpaste event? Yes, it's a thing, here's a JSFiddler so you can play with it yourself.

But there's also this regarding the event:

> *Non-standard event defined by Microsoft for use in Internet Explorer.*

C'mon guys, we've been down the non-standard implementations in browsers before and it always ends in tears.

Thing is though, not only does it work in IE but it works in Chrome too. And Safari. And Firefox. And even if it didn't, there are always bright JavaScript ideas to "hack" the ability of CTRL-V out of browsers.

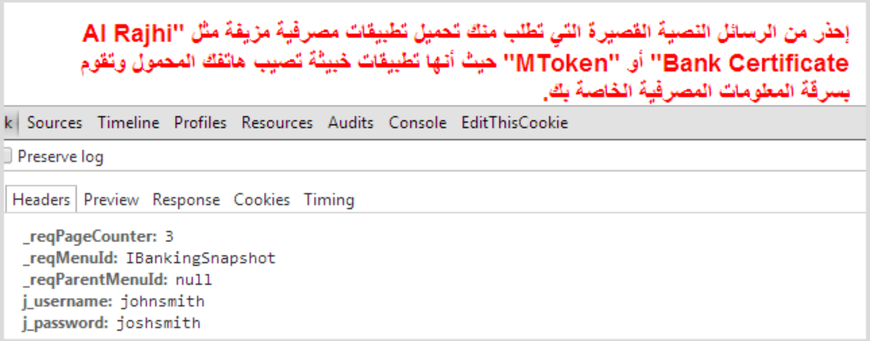
The point is that there are many ways of skinning this cat and it can be very easy. However, *it's a conscious decision* — the developer has to say "You know, I really don't think people need a paste facility; I'm gonna kill it".

But of course we really should touch on why people want to paste into password fields to begin with, so let's cover that now.

Why on earth would you want to paste a password anyway?!

The argument of whether you should use a password or a passphrase or just let the cat wander over the keyboard and see what happens has been well and truly had, and in the end it always comes back to this: The only secure password is the one you can't remember.

If you haven't already reached this enlightened state then free yourself from the shackles of human memory-based passwords and grab 1Password or KeePass or LastPass.



إحذر من الرسائل النصية القصيرة التي تطلب منك تحميل تطبيقات مصرفية مزيفة مثل "Al Rajhi
Bank Certificate" أو "MToken" حيث أنها تطبيقات خبيثة تصيب هاتفك المحمول وتقوم
بسرقة المعلومات المصرفية الخاصة بك.

```
_reqPageCounter: 3
_reqMenuId: IBankingSnapshot
_reqParentMenuId: null
j_username: johnsmith
j_password: joshsmith
```

▲ Figure 2: Error page from Al Rajhi bank

The premise of all of these is that you delegate "remembering" your password to the password manager and instead the only *real* memory required (at least of the human variety) is to remember the single master password that gains you access to all the other ones.

Again, the debates on the pros and cons of this have been had so go back to the aforementioned blog post and read the comments if you want to jump on that bandwagon.

Once a password manager has the credentials stored, clearly at some point in time, you'd want to get them back into a login form and actually use them to gain access to the website in question.

Password managers like 1Password can make this profoundly simple in that they have the ability to auto-fill the login form of the website in question. For example, I was able to successfully use it to attempt to login to the Al Rajhi bank (see figure 2).

As you can clearly see from the red error message, the login credentials were incorrect (I'm guessing "johnsmith" may not be that common of a name amongst their customers).

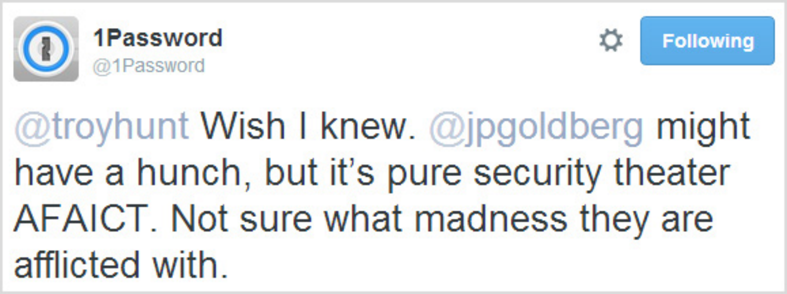
But as you can also clearly see from the request headers, both the username "johnsmith" and the password "joshsmith" were successfully sent. In other words, whilst non-standard onpaste implementation may be blocking the old CTRL-V, 1Password's ability to auto-fill the form is not hindered.

However, clever tricks like auto-fill don't always work. Sometimes the name of the form field changes either due to the site being revised or because of a perceived value of rotating and obfuscating the ID of the field.

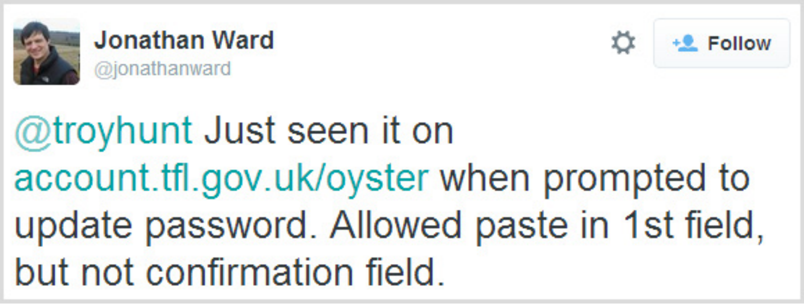
Sometimes you want to use the same credentials on multiple domains of the same service, and auto-fill only works against the domain the pattern was recorded on.

Sometimes you're in iOS and you really just want to use Safari, so you copy from the 1Password app and paste into the browser.

There are many, many valid reasons why people would want to paste passwords in order to *increase* their security profile, yet the perception of those blocking this practice is that it actually *decreases* security. Why? Interesting you should ask...



@troyhunt Wish I knew. @jpgoldberg might have a hunch, but it's pure security theater AFAICT. Not sure what madness they are afflicted with.



Jonathan Ward
@jonathanward

@troyhunt Just seen it on account.tfl.gov.uk/oyster when prompted to update password. Allowed paste in 1st field, but not confirmation field.

Because we've been bad, please create a weak password

I was admittedly mystified by this practice so I asked around and got a whole range of answers along the lines of "because some developers are just stupid".

Hard to argue with that although in fairness, this is often something dictated to them (although perhaps they should be doing a better job of articulating the counter-reasons).

I thought I'd ask 1Password, given these guys spend a bunch of time thinking about how to securely get creds into websites.

They pretty much concurred with everyone else (see tweet below).

Security theatre madness. Sounds about right.

But there's one angle to this that helps explain the madness and it goes back to that earlier PayPal screen grab.

This was of the *change password page*, not the login page. You can easily paste into the login page and in fact you can even paste into the *original password field* on the change password page, just not the new password field or the other field that confirms it.

But it's not just PayPal. Apparently it's the same with the Oyster Card site in the UK (see tweet).

You can easily paste into the password field of the login page for the Oyster card, but apparently you can't on the change password page. The reason lies in the earlier message I showed from PayPal, in particular this part of the password criteria:

> *Use 8-20 characters*

Ah, so because you've gone and put an arbitrary limit on the length of my password and taken away my ability to create a nice 50 character random string, you've had to kill the paste function, because otherwise I'd go around thinking I've got a 50 char password but it was actually truncated to 20 due to the maxlength attribute of the password field.

Nice one guys, good work there, bet you're saving a bundle of ingress data costs by cutting out those extra bytes!

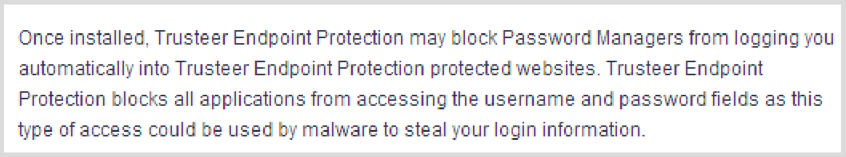
This, of course, is madness. Once passwords are stored, the hash of a 50 character password is an identical length to that of a 20 character one, which incidentally is also identical to that of a 100 character one. (Of course also keep in mind that there's a lot more to password storage than just hashing).

All of you creating these low arbitrary limits are actually hashing our passwords, right? Guys?

I've written about why we're forced into choosing bad passwords in the past and there's never really a good reason, just various shades of bad reasons.

That these bad reasons must then extend to disabling customers' ability to use a password manager to at least randomise the characters within the allowable limit only compounds the problem.

Here's an idea — if you



Once installed, Trusteer Endpoint Protection may block Password Managers from logging you automatically into Trusteer Endpoint Protection protected websites. Trusteer Endpoint Protection blocks all applications from accessing the username and password fields as this type of access could be used by malware to steal your login information.

▲ Figure 3: Message from Trusteer.com

really want to force people into creating short passwords because [insert baseless reason here], how about just watching for when the password field suddenly has the maximum number of allowable characters in it without the corresponding number of onkeypress events, and then saying "Excuse me, you've pasted a password we can't accept, could you please weaken it somewhat?"

But, but, but… malware could steal your login, yeah, that's it, malware!

There is a counter-argument to pasting passwords and it goes like this: if your computer is infected by malware, the nasties inside it may be able to access your clipboard and steal the password you copied onto it. Sound crazy? Trusteer doesn't think so. (see figure 3)

Of course the irony of this position is that it makes the assumption that a compromised machine may be at risk of its clipboard being accessed but not its keystrokes.

Why pull the password from memory for the small portion of people that elect to use a password manager when you can just grab the *keystrokes* with malware? Crikey, the operating system itself doesn't even need to be compromised to do that — you just grab a cheapie keylogger off the web and plug it into someone's USB keyboard. Job done.

In summary, just don't

Of course it all comes back to this balance in security where there are no absolutes and things are often a trade-off between cost, convenience and actually making the web a safer place.

But in the case of passwords, one of the best damn things anyone can do is get themselves a password manager and stop typing in that same crappy combination of kids' birthdays and dog's name they're using all over the place.

In the examples above, we've got a handful of websites forcing customers into creating arbitrarily short passwords, then disabling the ability to use password managers to the full extent possible. And to make it even worse, they're using non-standard browser behaviour to do it!

Hey, I've got an idea — maybe I should offer a bounty where if anyone comes to me and demonstrates that they have a crappy password on a site but is then able to kill it off and create a secure one, I'd give them some cash. What could go wrong?! ■

Reprinted with permission of the original author. First appeared at troyhunt.com.

Mental models I find repeatedly useful

By GABRIEL WEINBERG

A mental model is just a concept you can use to help try to explain things.

Around 2003, I came across Charlie Munger's 1995 speech, The Psychology of Human Misjudgment, which introduced me to how behavioral economics can be applied in business and investing. More profoundly, though, it opened my mind to the power of seeking out and applying mental models across a wide array of disciplines.

A mental model is just a concept you can use to help try to explain things (e.g. Hanlon's Razor—"never attribute to malice that which is adequately explained by carelessness.").

There are tens of thousands of mental models, and every discipline has their own set that you can learn through coursework, mentorship, or first-hand experience.

There is a much smaller set of concepts, however, that come up repeatedly in day-to-day decision making, problem solving, and truth seeking. As Munger says, "80 or 90 important models will carry about 90% of the freight in making you a worldly-wise person."

This post is my attempt to enumerate the mental models that are repeatedly useful to me. This set is clearly biased from my own experience and surely incomplete. I hope to continue to revise it as I remember and learn more.

How to use this list

I find mental models are useful to try to make sense of things and to help generate ideas. To actually be useful, however, you have to apply them in the right context at the right time. And for that to happen naturally, you have to know them well and practice using them.

Therefore, here are two suggestions for using this list:

1. For mental models you don't know or don't know well, you can use this list as a jumping-off point to study them. I've provided links (mainly to Wikipedia) to start that process.
2. When you have a particular problem in front of you, you can go down this list, and see if any of the models could possibly apply.

Notes

- Most of the mental models on this list are here because they are useful outside of their specific discipline. For example, use of the mental model "peak oil" isn't restricted to an energy context. Most references to "peak x" are an invocation of this model. Similarly, inflation as a concept applies outside of economics, e.g. grade inflation and expectations inflation.
- I roughly grouped the mental models by discipline, but as noted, this grouping is not to be taken as an assertion that they only apply within that discipline. The best ideas often arise when going cross-discipline.
- I realize my definition of mental model differs from some others, with mine being more broadly defined as any concept that helps explain, analyze, or navigate the world. I prefer this broader definition because it allows me to assemble a more wide-ranging list of useful concepts that may not be mental models under other definitions, but I nevertheless find on relatively equal footing in terms of usefulness in the real world.

If you're trying to be generally effective, my best advice is to start with the things on this list.

- The numbers next to each mental model reflect the frequency with which they come up:

  1. Frequently (63 models)
  2. Occasionally (43 models)
  3. Rarely, though still repeatedly (83 models)

- If studying new models, I'd start with the lower numbers first. The quotes next to each concept are meant to be a basic definition to remind you what it is, and is not a teaching tool. Follow the link to learn more.

- I am not endorsing any of these concepts as normatively good; I'm just saying they have repeatedly helped me explain and navigate the world.

- I wish I had learned many of these years earlier. In fact, the proximate cause for posting this was so I could more effectively answer the question I frequently get from people I work with: "What should I learn next?" If you're trying to be generally effective, my best advice is to start with the things on this list.

Explaining

- (1) Hanlon's razor—"Never attribute to malice that which is adequately explained by carelessness." (Related: fundamental attribution error—"the tendency for people to place an undue emphasis on internal characteristics of the agent (character or intention), rather than external factors, in explaining another person's behavior in a given situation.")
- (1) Occam's razor—"Among competing hypotheses, the one with the fewest assumptions should be selected." (Related: conjunction fallacy, overfitting, "when you hear hoof beats, think of horses not zebras.")
- (1) Cognitive biases—"Tendencies to think in certain ways that can lead to systematic deviations from a standard of rationality or good judgments." (See list of cognitive biases)
- (1) Arguing from first principles—"A first principle is a basic, foundational, self-evident proposition or assumption that cannot be deduced from any other proposition or assumption." (Related: dimensionality reduction, orthogonality)
- (1) Proximate vs root cause—"A proximate cause is an event which is closest to, or immediately responsible for causing, some observed result. This exists in contrast to a higher-level ultimate cause (or distal cause), which is usually thought of as the 'real' reason something occurred." (Related: 5 whys—"to determine the root cause of a defect or problem by repeating the question 'Why?')

Modeling

- (1) Systems thinking—"By taking the overall system as well as its parts into account, systems thinking is designed to avoid potentially contributing to further development of unintended consequences." (Related: causal loop diagrams, stock and flow, Le Chatelier's principle, hysteresis—"the time-based dependence of a system's output on present and past inputs.")
- (1) Scenario analysis—"A process of analyzing possi-

Pareto principle—"for many events, roughly 80% of the effects come from 20% of the causes."

ble future events by considering alternative possible outcomes."

- (1) Power-law—"A functional relationship between two quantities, where a relative change in one quantity results in a proportional relative change in the other quantity, independent of the initial size of those quantities: one quantity varies as a power of another." (Related: Pareto distribution; Pareto principle—"for many events, roughly 80% of the effects come from 20% of the causes.", diminishing returns, premature optimization, heavy-tailed distribution, fat-tailed distribution, long tail—"the portion of the distribution having a large number of occurrences far from the "head" or central part of the distribution."; black swan theory—"a metaphor that describes an event that comes as a surprise, has a major effect, and is often inappropriately rationalized after the fact with the benefit of hindsight.")
- (1) Normal distribution—"A very common continuous probability distribution… physical quantities that are expected to be the sum of many independent processes (such as measurement errors) often have distributions that are nearly normal." (Related: central limit theorem)
- (1) Sensitivity analysis—"The study of how the uncertainty in the output of a mathematical model or system (numerical or otherwise) can be apportioned to different sources of uncertainty in its inputs."
- (1) Cost-benefit analysis—"A systematic approach to estimating the strengths and weaknesses of alternatives that satisfy transactions, activities or functional requirements for a business." (Related: net present value—"a measurement of the profitability of an undertaking that is calculated by subtracting the present values of cash outflows (including initial cost) from the present values of cash inflows over a period of time.", discount rate)
- (3) Simulation—"The imitation of the operation of a real-world process or system over time."
- (3) Pareto efficiency—"A state of allocation of resources in which it is impossible to make any one individual better off without making at least one individual worse off…A Pareto improvement is defined to be a change to a different allocation that makes at least one individual better off without making any other individual worse off, given a certain initial allocation of goods among a set of individuals."

Brainstorming

- (1) Lateral thinking—"Solving problems through an indirect and creative approach, using reasoning that is not immediately obvious and involving ideas that may not be obtainable by using only traditional step-by-step logic."
- (1) Divergent thinking vs convergent thinking—"Divergent thinking is a thought process or method used to generate creative ideas by exploring many possible solutions. It is often used in conjunction with its cognitive opposite, conver-

> ***Leverage—“The force amplification achieved by using a tool, mechanical device or machine system.”***

gent thinking, which follows a particular set of logical steps to arrive at one solution, which in some cases is a ‘correct’ solution.” (Related: groupthink)

- (2) Critical mass—“The smallest amount of fissile material needed for a sustained nuclear chain reaction.” “In social dynamics, critical mass is a sufficient number of adopters of an innovation in a social system so that the rate of adoption becomes self-sustaining and creates further growth.”
- (2) Activation energy—“The minimum energy that must be available to a chemical system with potential reactants to result in a chemical reaction.”
- (2) Catalyst—“A substance that increases the rate of a chemical reaction.”
- (2) Leverage—“The force amplification achieved by using a tool, mechanical device or machine system.”
- (2) Crowdsourcing—“The process of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, especially an online community, rather than from employees or suppliers.” (Related: wisdom of the crowd—“a large group’s aggregated answers to questions involving quantity estimation, general world knowledge, and spatial reasoning has generally been found to be as good as, and often better than, the answer given by any of the individuals within the group.”, collective intelligence)
- (3) The structure of scientific revolutions—“An episodic model in which periods of such conceptual continuity in normal science were interrupted by periods of revolutionary science. The discovery of “anomalies” during revolutions in science leads to new paradigms. New paradigms then ask new questions of old data, move beyond the mere “puzzle-solving” of the previous paradigm, change the rules of the game and the “map” directing new research.” (Related: Planck’s principle—“the view that scientific change does not occur because individual scientists change their mind, but rather that successive generations of scientists have different views.”)

Experimenting

- (1) Scientific method—“Systematic observation, measurement, and experiment, and the formulation, testing, and modification of hypotheses.” (Related: reproducibility)
- (1) Proxy—“A variable that is not in itself directly relevant, but that serves in place of an unobservable or immeasurable variable. In order for a variable to be a good proxy, it must have a close correlation, not necessarily linear, with the variable of interest.” (Related: revealed preference)
- (1) Selection bias—“The selection of individuals, groups or data for analysis in such a way that proper randomization is not achieved, thereby ensuring that the sample obtained is not representative of the population intended to be analyzed.” (Related: sampling bias)
- (1) Response bias—“A wide range of cognitive

Survivorship bias—“The logical error of concentrating on the people or things that ‘survived’ and inadvertently overlooking those that did not.”

biases that influence the responses of participants away from an accurate or truthful response.”

- (2) Observer effect—“Changes that the act of observation will make on a phenomenon being observed.” (Related: Schrödinger’s cat)
- (2) Survivorship bias—“The logical error of concentrating on the people or things that ‘survived’ some process and inadvertently overlooking those that did not because of their lack of visibility.”
- (3) Heisenberg uncertainty principle—“A fundamental limit to the precision with which certain pairs of physical properties of a particle, known as complementary variables, such as position *x* and momentum *p*, can be known.”

Interpreting

- (1) Order of magnitude—“An order-of-magnitude estimate of a variable whose precise value is unknown is an estimate rounded to the nearest power of ten.” (Related: order of approximation, back-of-the-envelope calculation, dimensional analysis, Fermi problem)
- (1) Major vs minor factors—Major factors explains major portions of the results, while minor factors only explain minor portions. (Related: first order vs second order effects—first order effects directly follow from a cause, while second order effects follow from first order effects.)
- (1) False positives and false negatives—“A false positive error, or in short false positive, commonly called a ‘false alarm’, is a result that indicates a given condition has been fulfilled, when it actually has not been fulfilled…A false negative error, or in short false negative, is where a test result indicates that a condition failed, while it actually was successful, i.e. erroneously no effect has been assumed.”
- (1) Confidence interval—“Confidence intervals consist of a range of values (interval) that act as good estimates of the unknown population parameter; however, the interval computed from a particular sample does not necessarily include the true value of the parameter.” (Related: error bar)
- (2) Bayes’ theorem—“Describes the probability of an event, based on conditions that might be related to the event. For example, suppose one is interested in whether a person has cancer, and knows the person’s age. If cancer is related to age, then, using Bayes’ theorem, information about the person’s age can be used to more accurately assess the probability that they have cancer.” (Related: base rate fallacy)
- (2) Regression to the mean—“The phenomenon that if a variable is extreme on its first measurement, it will tend to be closer to the average on its second measurement.”
- (2) Inflection point—“A point on a curve at which the curve changes from being concave (concave downward) to convex (concave upward), or vice versa.”
- (3) Simpson’s paradox—“A paradox in probability and statistics, in which a trend

> ***False cause—“Presuming that a real or perceived relationship between things means that one is the cause of the other.”***

appears in different groups of data but disappears or reverses when these groups are combined.”

Deciding

- (1) Business case—“Captures the reasoning for initiating a project or task. It is often presented in a well-structured written document, but may also sometimes come in the form of a short verbal argument or presentation.” (Related: why this now?)
- (1) Opportunity cost—“The value of the best alternative forgone where, given limited resources, a choice needs to be made between several mutually exclusive alternatives. Assuming the best choice is made, it is the ‘cost’ incurred by not enjoying the benefit that would have been had by taking the second best available choice.” (Related: cost of capital)
- (1) Intuition—Personal experience coded into your personal neural network, which means your intuition is dangerous outside the bounds of your personal experience. (Related: thinking fast vs thinking slow—“a dichotomy between two modes of thought: ‘System 1’ is fast, instinctive and emotional; ‘System 2’ is slower, more deliberative, and more logical.”)
- (1) Local vs global optimum—“A local optimum of an optimization problem is a solution that is optimal (either maximal or minimal) within a neighboring set of candidate solutions. This is in contrast to a global optimum, which is the optimal solution among all possible solutions, not just those in a particular neighborhood of values.”
- (1) Decision trees—“A decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.” (Related: expected value)
- (1) Sunk cost—“A cost that has already been incurred and cannot be recovered.” (Related: “throwing good money after bad”, “in for a penny, in for a pound”)
- (1) Availability bias—“People tend to heavily weigh their judgments toward more recent information, making new opinions biased toward that latest news.”
- (1) Confirmation bias—“The tendency to search for, interpret, favor, and recall information in a way that confirms one’s preexisting beliefs or hypotheses, while giving disproportionately less consideration to alternative possibilities.” (Related: cognitive dissonance)
- (3) Loss aversion—“People’s tendency to strongly prefer avoiding losses to acquiring gains.” (Related: diminishing marginal utility)

Reasoning

- (1) Anecdotal—“Using a personal experience or an isolated example instead of a sound argument or compelling evidence.”
- (1) False cause—“Presuming that a real or perceived relationship between things means that one is the cause of the other.” (Related: correlation does not imply causation, or in xkcd form)

Likely—Thinking that just because something is possible means that it is likely.

- (1) Straw man—“Giving the impression of refuting an opponent’s argument, while actually refuting an argument that was not advanced by that opponent.”
- (1) Plausible—Thinking that just because something is plausible means that it is true.
- (1) Likely—Thinking that just because something is possible means that it is likely.
- (1) Appeal to emotion—“Manipulating an emotional response in place of a valid or compelling argument.”
- (1) Ad hominem—“Attacking your opponent’s character or personal traits in an attempt to undermine their argument.”
- (1) Slippery slope—“Asserting that if we allow A to happen, then Z will eventually happen too, therefore A should not happen.” (Related: broken windows theory—“maintaining and monitoring urban environments to prevent small crimes such as vandalism, public drinking, and toll-jumping helps to create an atmosphere of order and lawfulness, thereby preventing more serious crimes from happening.”)
- (1) Black or white—“When two alternative states are presented as the only possibilities, when in fact more possibilities exist.”
- (1) Bandwagon—“Appealing to popularity or the fact that many people do something as an attempted form of validation.”
- For a longer list, see Thou shall not commit logical fallacies (I have this poster on my office door.)

Negotiating

- (1) The third story—“The third story is one an impartial observer, such as a mediator, would tell; it’s a version of events both sides can agree on.”
- (1) Active listening—“Requires that the listener fully concentrates, understands, responds and then remembers what is being said.”
- (1) Trade-offs—“A situation that involves losing one quality or aspect of something in return for gaining another quality or aspect.”
- (1) Incentives—“Something that motivates an individual to perform an action.”
- (2) Best alternative to a negotiated agreement (BATNA)—“The most advantageous alternative course of action a party can take if negotiations fail and an agreement cannot be reached.”
- (2) Zero-sum vs non-zero-sum—“A zero-sum game is a mathematical representation of a situation in which each participant’s gain (or loss) of utility is exactly balanced by the losses (or gains) of the utility of the other participant(s)…In contrast, non-zero-sum describes a situation in which the interacting parties’ aggregate gains and losses can be less than or more than zero.” (Related: win-win—“A win–win strategy is a conflict resolution process that aims to accommodate all disputants.”
- (3) Alternative dispute resolution (ADR)—“Dispute resolution processes and techniques that act as a means for disagreeing parties to come to an agreement short of litigation.” (Related: mediation, arbitration)

Goodhart's law—"When a measure becomes a target, it ceases to be a good measure."

- (3) Prisoner's dilemma—"A standard example of a game analyzed in game theory that shows why two completely 'rational' individuals might not cooperate, even if it appears that it is in their best interests to do so." (Related: Nash equilibrium, evolutionarily stable strategy)

Mitigating

- (1) Unintended consequences—"Outcomes that are not the ones foreseen and intended by a purposeful action." (Related: collateral damage—"Deaths, injuries, or other damage inflicted on an unintended target.", Goodhart's law—"When a measure becomes a target, it ceases to be a good measure", Campbell's law, Streisand effect—"The phenomenon whereby an attempt to hide, remove, or censor a piece of information has the unintended consequence of publicizing the information more widely, usually facilitated by the Internet", cobra effect—"when an attempted solution to a problem actually makes the problem worse.")
- (2) Preserving optionality—"A strategy of keeping options open and fluid, fighting the urge to make choices too soon, before all of the uncertainties have been resolved." (Related: tyranny of small decisions—"a situation where a series of small, individually rational decisions can negatively change the context of subsequent choices, even to the point where desired alternatives are irreversibly destroyed.", boiling frog—"an anecdote describing a frog slowly being boiled alive.", path dependence)
- (2) Precautionary principle—"If an action or policy has a suspected risk of causing harm to the public, or to the environment, in the absence of scientific consensus (that the action or policy is not harmful), the burden of proof that it is not harmful falls on those taking an action that may or may not be a risk."
- (2) Short-termism—"Short-termism refers to an excessive focus on short-term results at the expense of long-term interests."

Managing

- (1) Weekly 1-1s—"1-1's can add a whole new level of speed and agility to your company."
- (1) Forcing function—"A forcing function is any task, activity or event that forces you to take action and produce a result."
- (1) Pygmalion effect—"The phenomenon whereby higher expectations lead to an increase in performance." (Related: market pull technology policy—where the government sets future standards beyond what the current market can deliver, and the market pulls that technology into existence.)
- (1) Virtual team—"A group of individuals who work across time, space and organizational boundaries with links strengthened by webs of communication technology." At least in some circumstances, it is possible to have a completely virtual team. The downsides in lack of face-to-face communication can be outweighed by the upsides in sourcing from the entire world.

> ***Hedgehog vs fox—“A fox knows many things, but a hedgehog knows one important thing.”***

- (2) Introversion vs extraversion—“Extraversion tends to be manifested in outgoing, talkative, energetic behavior, whereas introversion is manifested in more reserved and solitary behavior. Virtually all comprehensive models of personality include these concepts in various forms.”
- (2) IQ vs EQ—“IQ is a total score derived from one of several standardized tests designed to assess human intelligence.” “EQ is the capacity of individuals to recognize their own, and other people’s emotions, to discriminate between different feelings and label them appropriately, and to use emotional information to guide thinking and behavior.”
- (2) Growth mindset vs fixed mindset—“Those with a ‘fixed mindset’ believe that abilities are mostly innate and interpret failure as the lack of necessary basic abilities, while those with a ‘growth mindset’ believe that they can acquire any given ability provided they invest effort or study.”
- (2) Hindsight bias—“The inclination, after an event has occurred, to see the event as having been predictable, despite there having been little or no objective basis for predicting it.” (Related: Pollyanna principle—“tendency for people to remember pleasant items more accurately than unpleasant ones”)
- (2) Organizational debt—“All the people/culture compromises made to ‘just get it done’ in the early stages of a startup.”
- (2) Generalist vs specialist—“A generalist is a person with a wide array of knowledge, the opposite of which is a specialist.” (Related: hedgehog vs fox—“A fox knows many things, but a hedgehog knows one important thing.”)
- (2) Consequence vs conviction—“Where there is low consequence and you have very low confidence in your own opinion, you should absolutely delegate. And delegate completely, let people make mistakes and learn. On the other side, obviously where the consequences are dramatic and you have extremely high conviction that you are right, you actually can’t let your junior colleague make a mistake.”
- (3) High-context vs low-context culture—“In a higher-context culture, many things are left unsaid, letting the culture explain. Words and word choice become very important in higher-context communication, since a few words can communicate a complex message very effectively to an in-group (but less effectively outside that group), while in a low-context culture, the communicator needs to be much more explicit and the value of a single word is less important.”
- (3) Peter principle—“The selection of a candidate for a position is based on the candidate’s performance in their current role, rather than on abilities relevant to the intended role. Thus, employees only stop being promoted once they can no longer perform effectively, and ‘managers rise to the level of their incompetence.’”
- (3) Maslow’s hierarchy of needs—“Maslow used the terms ‘physiological’, ‘safe-

Zawinski's law—"Every program attempts to expand until it can read mail. Those programs which cannot so expand are replaced by ones which can."

ty', 'belongingness' and 'love', 'esteem', 'self-actualization', and 'self-transcendence' to describe the pattern that human motivations generally move through…[though there is] little evidence for the ranking of needs that Maslow described or for the existence of a definite hierarchy at all."

- (3) Loyalists vs mercenaries—"There are highly loyal teams that can withstand almost anything and remain steadfastly behind their leader. And there are teams that are entirely mercenary and will walk out without thinking twice about it."
- (3) Dunbar's number—"A suggested cognitive limit to the number of people with whom one can maintain stable social relationships…with a commonly used value of 150."
- (3) Zero tolerance—"Strict punishment for infractions of a stated rule, with the intention of eliminating undesirable conduct."
- (3) Commandos vs Infantry vs Police—"Three distinct groups of people that define the lifetime of a company: Commandos, Infantry, and Police: Whether invading countries or markets, the first wave of troops to see battle are the commandos… Grouping offshore as the commandos do their work is the second wave of soldiers, the infantry…But there is still a need for a military presence in the territory they leave behind, which they have liberated. These third-wave troops hate change. They aren't troops at all but police."

Developing

- (1) Technical debt—"A concept in programming that reflects the extra development work that arises when code that is easy to implement in the short run is used instead of applying the best overall solution."
- (1) Binary search—"A search algorithm that finds the position of a target value within a sorted array. It compares the target value to the middle element of the array; if they are unequal, the half in which the target cannot lie is eliminated and the search continues on the remaining half until it is successful." (Related: debugging)
- (1) Divide and conquer—"Recursively breaking down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem."
- (1) Design pattern—"The re-usable form of a solution to a design problem." (Related: anti-pattern—"a common response to a recurring problem that is usually ineffective and risks being highly counterproductive.", dark pattern—"user interfaces designed to trick people.")
- (3) Zawinski's law—"Every program attempts to expand until it can read mail. Those programs which cannot so expand are replaced by ones which can." (Related: Greenspun's tenth rule—"any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp.")

Clarke’s third law—“Any sufficiently advanced technology is indistinguishable from magic.”

- (3) Moore’s law—“The observation that the number of transistors in a dense integrated circuit doubles approximately every two years.”
- (3) Metcalfe's Law—“The value of a telecommunications network is proportional to the square of the number of connected users of the system…Within the context of social networks, many, including Metcalfe himself, have proposed modified models using (*n* × log *n*) proportionality rather than *n^2* proportionality.”
- (3) Clarke’s third law—“Any sufficiently advanced technology is indistinguishable from magic.”

Business

- (1) Minimum viable product—“A product with just enough features to gather validated learning about the product and its continued development.” (Related: perfect is the enemy of good)
- (2) Capital allocation options—“Five capital allocation choices CEOs have: 1) invest in existing operations; 2) acquire other businesses; 3) issue dividends; 4) pay down debt; 5) repurchase stock. Along with this, they have three means of generating capital: 1) internal/ operational cash flow; 2) debt issuance; 3) equity issuance.”
- (3) Luck surface area—“When you do something you’re excited about you will naturally pull others into your orbit. And the more people with whom you share your passion, the more who will be pulled into your orbit.”
- (3) Hunting elephants vs flies—“Salespeople sometimes refer to ‘elephants’, ‘deers’ and ‘rabbits’ when they talk about the first three categories of customers. To extend the metaphor to the 4th and 5th type of customer, let’s call them ‘mice” and “flies’. So how can you hunt 1,000 elephants, 10,000 deer, 100,000 rabbits, 1,000,000 mice or 10,000,000 flies?” (Related: brontosaurus, whale, and microbe)
- (3) Secrets—“Every one of today’s most famous and familiar ideas was once unknown and unsuspected…There are many more secrets left to find, but they will yield only to relentless searchers.”
- (3) Strategic acquisition vs financial acquisition vs acquihire—Different motivations for an acquiring company typically have significantly different valuation models. (Related: rollup—“a technique used by investors (commonly private equity firms) where multiple small companies in the same market are acquired and merged.”, P/E-driven acquisitions, auction)

Influencing

- (1) Framing—“With the same information being used as a base, the ‘frame’ surrounding the issue can change the reader’s perception without having to alter the actual facts.” (Related: anchoring)
- (2) Cialdini’s six principles of influence—Reciprocity (“People tend to return a favor.”), Commitment (“If people commit…they are more likely to honor that commit-

Coda—"A term used in music primarily to designated a passage that brings a piece to an end."

ment."), Social Proof ("People will do things they see other people are doing."), Authority ("People will tend to obey authority figures."), Liking ("People are easily persuaded by other people they like."), and Scarcity ("Perceived scarcity will generate demand"). (Related: foot-in-the-door technique)

- (3) Paradox of choice—"Eliminating consumer choices can greatly reduce anxiety for shoppers." (Related: Hick's law, "increasing the number of choices will increase the decision time logarithmically.")
- (3) Major vs minor chords—"In Western music, a minor chord, in comparison, 'sounds darker than a major chord.'"
- (3) Coda—"A term used in music primarily to designated a passage that brings a piece to an end." (Related: CTA.) People psychologically expect codas, and so they can be used for influence.

Marketing

- (1) Bullseye framework—"With nineteen traction channels to consider, figuring out which one to focus on is tough. That's why we've created a simple framework called Bullseye that will help you find *the channel* that will get you traction."
- (1) Technology adoption lifecycle—"Describes the adoption or acceptance of a new product or innovation, according to the demographic and psychological characteristics of defined adopter groups. The process of adoption over time is typically illustrated as a classical normal distribution or "bell curve". The model indicates that the first group of people to use a new product is called 'innovators', followed by 'early adopters'. Next come the early majority and late majority, and the last group to eventually adopt a product are called 'laggards'." (Related: S-curve, crossing the chasm, installation period vs deployment period)
- (3) Jobs to be done—"Consumers usually don't go about their shopping by conforming to particular segments. Rather, they take life as it comes. And when faced with a job that needs doing, they essentially 'hire' a product to do that job."
- (3) Fear, uncertainty, and doubt (FUD)—"A disinformation strategy used in sales, marketing, public relations, politics and propaganda. FUD is generally a strategy to influence perception by disseminating negative and dubious or false information, and a manifestation of the appeal to fear."

Competing

- (2) Supply and demand—"An economic model of price determination in a market. It concludes that in a competitive market, the unit price for a particular good, or other traded item such as labor or liquid financial assets, will vary until it settles at a point where the quantity demanded (at the current price) will equal the quantity supplied (at the current price), resulting in an economic equilib-

> Giffen good—“a product that people consume more of as the price rises and vice versa.”

rium for price and quantity transacted.” (Related: perfect competition, arbitrage—“the practice of taking advantage of a price difference between two or more markets.”)

- (2) Winner take all market—A market that tends towards one dominant player. (Related: lock-in, monopoly, monopsony)
- (2) Two-sided market—“Economic platforms having two distinct user groups that provide each other with network benefits.”
- (3) Barriers to entry—“A cost that must be incurred by a new entrant into a market that incumbents don’t or haven’t had to incur.”
- (3) Price elasticity—“The measurement of how responsive an economic variable is to a change in another. It gives answers to questions such as ‘If I lower the price of a product, how much more will sell?’” (Related: Giffen good—“a product that people consume more of as the price rises and vice versa.”)
- (3) Market power—“The ability of a firm to profitably raise the market price of a good or service over marginal cost.”
- (3) Conspicuous consumption—“The spending of money on and the acquiring of luxury goods and services to publicly display economic power.” (Related: Veblen goods—“types of luxury goods, such as expensive wines, jewelry, fashion-designer handbags, and luxury cars, which are in demand because of the high prices asked for them.”)
- (3) Comparative advantage—“An agent has a comparative advantage over another in producing a particular good if they can produce that good at a lower relative opportunity cost or autarky price, i.e. at a lower relative marginal cost prior to trade.”
- (3) Creative destruction—“Process of industrial mutation that incessantly revolutionizes the economic structure from within, incessantly destroying the old one, incessantly creating a new one.”

Strategizing

- (1) Sustainable competitive advantage—Structural factors that allow a firm to outcompete its rivals for many years.
- (1) Core competency—“A harmonized combination of multiple resources and skills that distinguish a firm in the marketplace.” (Related: circle of competence—“you don’t have to be an expert on every company, or even many. You only have to be able to evaluate companies within your circle of competence. The size of that circle is not very important; knowing its boundaries, however, is vital.”)
- (1) Strategy vs tactics—Sun Tzu: “Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat.”
- (1) Sphere of influence—“A spatial region or concept division over which a state or organization has a level of cultural, economic, military, or political exclusivity, accommodating to the interests of powers outside the borders of the state that controls it.”

Switching costs—“The costs associated with switching suppliers.”

- (2) Unknown unknowns—“Known unknowns refers to ‘risks you are aware of, such as cancelled flights…’ Unknown unknowns are risks that ‘come from situations that are so out of this world that they never occur to you.’
- (2) Switching costs—“The costs associated with switching suppliers.”
- (3) Network effect—“The effect that one user of a good or service has on the value of that product to other people. When a network effect is present, the value of a product or service is dependent on the number of others using it.”
- (3) Economies of scale—“The cost advantages that enterprises obtain due to size, output, or scale of operation, with cost per unit of output generally decreasing with increasing scale as fixed costs are spread out over more units of output.”

Military

- (3) Two-front war— “A war in which fighting takes place on two geographically separate fronts.”
- (3) Flypaper theory— “The idea that it is desirable to draw enemies to a single area, where it is easier to kill them and they are far from one’s own vulnerabilities.” (Related: honeypot)
- (3) Fighting the last war— Using strategies and tactics that worked successfully in the past, but are no longer as useful.
- (3) Rumsfeld’s rule— “You go to war with the Army you have. They’re not the Army you might want or wish to have at a later time.” (Related: Joy’s law— “no matter who you are, most of the smartest people work for someone else.”)
- (3) Trojan horse— “After a fruitless 10-year siege, the Greeks constructed a huge wooden horse, and hid a select force of men inside. The Greeks pretended to sail away, and the Trojans pulled the horse into their city as a victory trophy. That night the Greek force crept out of the horse and opened the gates for the rest of the Greek army, which had sailed back under cover of night. The Greeks entered and destroyed.”
- (3) Empty fort strategy— “Involves using reverse psychology (and luck) to deceive the enemy into thinking that an empty location is full of traps and ambushes, and therefore induce the enemy to retreat.” (Related: vaporware— “a product, typically computer hardware or software, that is announced to the general public but is never actually manufactured nor officially cancelled.”
- (3) Exit strategy— “A means of leaving one’s current situation, either after a predetermined objective has been achieved, or as a strategy to mitigate failure.”
- (3) Boots on the ground— “The belief that military success can only be achieved through the direct physical presence of troops in a conflict area.”
- (3) Winning hearts and minds— “In which one side seeks to prevail not by the use of superior force, but by making emotional or intellectual appeals to sway supporters of the other side.”

Information asymmetry—"The study of decisions in transactions where one party has more or better information than the other."

- (3) Mutually assured destruction—"In which a full-scale use of nuclear weapons by two or more opposing sides would cause the complete annihilation of both the attacker and the defender. It is based on the theory of deterrence, which holds that the threat of using strong weapons against the enemy prevents." (Related: Mexican standoff, Zugzwang)
- (3) Containment— "A military strategy to stop the expansion of an enemy. It is best known as the Cold War policy of the United States and its allies to prevent the spread of communism abroad."
- (3) Appeasement—"A diplomatic policy of making political or material concessions to an enemy power in order to avoid conflict." (Related: Danegeld, extortion)
- (3) Winning a battle but losing the war—"A poor strategy that wins a lesser (or sub-) objective but overlooks and loses the true intended objective." (Related: sacrifice play)
- (3) Beachhead— "A temporary line created when a military unit reaches a landing beach by sea and begins to defend the area while other reinforcements help out until a unit large enough to begin advancing has arrived."
- (3) Proxy war—"A conflict between two nations where neither country directly engages the other."

Sports

- (3) Hail Mary pass—"A very long forward pass in American football, made in desperation with only a small chance of success… has become generalized to refer to any last-ditch effort with little chance of success."

Market failure

- (2) Social vs market norms—"People are happy to do things occasionally when they are not paid for them. In fact there are some situations in which work output is negatively affected by payment of small amounts of money."
- (3) Information asymmetry—"The study of decisions in transactions where one party has more or better information than the other." (Related: adverse selection—"when traders with better private information about the quality of a product will selectively participate in trades which benefit them the most"; moral hazard—"when one person takes more risks because someone else bears the cost of those risks.")
- (3) Externalities—"An externality is the cost or benefit that affects a party who did not choose to incur that cost or benefit." (Related: tragedy of the commons—"A situation within a shared-resource system where individual users acting independently according to their own self-interest behave contrary to the common good of all users by depleting that resource through their collective action"; free rider problem—"when those who benefit from resources, goods, or services do not pay for them, which results in an under-provision of those goods or services."; Coase theorem—"if trade in an externality is

Shirky principle—"Institutions will try to preserve the problem to which they are the solution."

possible and there are sufficiently low transaction costs, bargaining will lead to a Pareto efficient outcome regardless of the initial allocation of property.")

- (3) Deadweight loss—"A loss of economic efficiency that can occur when equilibrium for a good or service is not achieved or is not achievable."

Political failure

- (2) Chilling effect—"The inhibition or discouragement of the legitimate exercise of natural and legal rights by the threat of legal sanction… Outside of the legal context in common usage; any coercion or threat of coercion (or other unpleasantries) can have a chilling effect on a group of people regarding a specific behavior, and often can be statistically measured or be plainly observed."
- (3) Regulatory capture—"When a regulatory agency, created to act in the public interest, instead advances the commercial or political concerns of special interest groups that dominate the industry or sector it is charged with regulating." (Related: Shirky principle—"Institutions will try to preserve the problem to which they are the solution.")
- (3) Duverger's law—"A principle which states that plurality-rule elections (such as *first past the post*) structured within single-member districts tend to favor a two-party system, and that 'the double ballot majority system and proportional representation tend to favor multipartism.'"
- (3) Arrow's impossibility theorem—"When voters have three or more distinct alternatives (options), no ranked order voting system can convert the ranked preferences of individuals into a community-wide (complete and transitive) ranking while also meeting a pre-specified set of criteria." (Related: approval voting)

Investing

- (2) Fear of missing out (FOMO)—"A pervasive apprehension that others might be having rewarding experiences from which one is absent."
- (2) Preferred stock vs common stock—"Preferred stock is a type of stock which may have any combination of features not possessed by common stock including properties of both an equity and a debt instrument, and is generally considered a hybrid instrument."
- (3) Margin of safety— "The difference between the intrinsic value of a stock and its market price."
- (3) Investing vs speculation—"Typically, high-risk trades that are almost akin to gambling fall under the umbrella of speculation, whereas lower-risk investments based on fundamentals and analysis fall into the category of investing."
- (3) Compound interest—"Interest on interest. It is the result of reinvesting interest, rather than paying it out, so that interest in the next period is then earned on the principal sum plus previously-accumulated interest."
- (3) Inflation—"A sustained increase in the general price

> ***Makers vs manager's schedule—"When you're operating on the maker's schedule, meetings are a disaster."***

level of goods and services in an economy over a period of time." (Related: real vs nominal value, hyperinflation, deflation, debasement)

- (3) Gross domestic product (GDP)—"A monetary measure of the market value of all final goods and services produced in a period (quarterly or yearly)."
- (3) Efficient-market hypothesis—"Asset prices fully reflect all available information…Investors, including the likes of Warren Buffett, and researchers have disputed the efficient-market hypothesis both empirically and theoretically." (Related: alpha)
- (3) Purchasing power parity—"Allows one to estimate what the exchange rate between two currencies would have to be in order for the exchange to be at par with the purchasing power of the two countries' currencies."
- (3) Insider trading—"The trading of a public company's stock or other securities (such as bonds or stock options) by individuals with access to nonpublic information about the company."
- (3) Poison pill—"A type of defensive tactic used by a corporation's board of directors against a takeover. Typically, such a plan gives shareholders the right to buy more shares at a discount if one shareholder buys a certain percentage or more of the company's shares." (Related: proxy fight).

Learning

- (1) Deliberate practice—"How expert one becomes at a skill has more to do with how one practices than with merely performing a skill a large number of times."
- (3) Imposter syndrome—"High-achieving individuals marked by an inability to internalize their accomplishments and a persistent fear of being exposed as a 'fraud'."
- (3) Dunning-Kruger effect—"Relatively unskilled persons suffer illusory superiority, mistakenly assessing their ability to be much higher than it really is…[and] highly skilled individuals may underestimate their relative competence and may erroneously assume that tasks which are easy for them are also easy for others." (Related: overconfidence effect)
- (3) Spacing effect—"The phenomenon whereby learning is greater when studying is spread out over time, as opposed to studying the same amount of time in a single session."

Productivity

- (1) Focus on high-leverage activities—"Leverage should be the central, guiding metric that helps you determine where to focus your time." (Related: Eisenhower decision matrix—"what is important is seldom urgent, and what is urgent is seldom important.", "The best time to plant a tree was 20 years ago. The second best time is now.", law of triviality—"members of an organisation give disproportionate weight to trivial issues.")
- (1) Makers vs manager's schedule—"When you're operating on the maker's schedule, meetings are a disaster." (Related: deep work)

Gate's law—"Most people overestimate what they can do in one year and underestimate what they can do in ten years."

- (2) Murphy's law—"Anything that can go wrong, will." (Related:Hofstadter's law, "It always takes longer than you expect, even when you take into account Hofstadter's law.")
- (3) Parkinson's law—"Work expands so as to fill the time available for its completion."
- (3) Gate's law—"Most people overestimate what they can do in one year and underestimate what they can do in ten years."

Nature

- (2) Nature vs nurture—"the relative importance of an individual's innate qualities as compared to an individual's personal experiences in causing individual differences, especially in behavioral traits."
- (2) Chain reaction—"A sequence of reactions where a reactive product or by-product causes additional reactions to take place. In a chain reaction, positive feedback leads to a self-amplifying chain of events." (Related: cascading failure, domino effect)
- (2) Filling a vacuum—A vacuum "is space void of matter." Filling a vacuum refers to the fact that if a vacuum is put next to something with pressure, it will be quickly filled by the gas producing that pressure. (Related: power vacuum)
- (2) Emergence—"Whereby larger entities, patterns, and regularities arise through interactions among smaller or simpler entities that themselves do not exhibit such properties." (Related: decentralized system, spontaneous order)
- (3) Natural selection— "The differential survival and reproduction of individuals due to differences in phenotype. It is a key mechanism of evolution, the change in heritable traits of a population over time."
- (3) Butterfly effect—"The concept that small causes can have large effects." (Related: bullwhip effect—"increasing swings in inventory in response to shifts in customer demand as you move further up the supply chain.")
- (3) Sustainability— "The endurance of systems and processes."
- (3) Peak oil— "The point in time when the maximum rate of extraction of petroleum is reached, after which it is expected to enter terminal decline."

Philosophy

- (2) Consequentialism—"Holding that the consequences of one's conduct are the ultimate basis for any judgment about the rightness or wrongness of that conduct." (Related: "ends justify the means")
- (2) Distributive justice vs procedural justice—"Procedural justice concerns the fairness and the transparency of the processes by which decisions are made, and may be contrasted with distributive justice (fairness in the distribution of rights or resources), and retributive justice (fairness in the punishment of wrongs)."
- (3) Effective altruism—"Encourages individuals to consider all causes and actions, and then act in the way that brings about the greatest

Utilitarianism—"Holding that the best moral action is the one that maximizes utility."

positive impact, based on their values."

- (3) Utilitarianism—"Holding that the best moral action is the one that maximizes utility."
- (3) Agnosticism—"The view that the truth values of certain claims—especially metaphysical and religious claims such as whether God, the divine, or the supernatural exist—are unknown and perhaps unknowable."
- (3) Veil of ignorance—"A method of determining the morality of a certain issue (e.g., slavery) based upon the following thought experiment: parties to the original position know nothing about the particular abilities, tastes, and positions individuals will have within a social order. When such parties are selecting the principles for distribution of rights, positions, and resources in the society in which they will live, the veil of ignorance prevents them from knowing who will receive a given distribution of rights, positions, and resources in that society."

Internet

- (2) Filter bubble—"In which a website algorithm selectively guesses what information a user would like to see based on information about the user (such as location, past click behavior and search history) and, as a result, users become separated from information that disagrees with their viewpoints, effectively isolating them in their own cultural or ideological bubbles." (Related: echo chamber)
- (2) Botnet—"A number of Internet-connected computers communicating with other similar machines in which components located on networked computers communicate and coordinate their actions by command and control (C&C) or by passing messages to one another." (Related: flash mob)
- (2) Spamming—"The use of electronic messaging systems to send unsolicited messages (spam), especially advertising, as well as sending messages repeatedly on the same site." (Related: phishing—"the attempt to acquire sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a trustworthy entity in an electronic communication.", clickjacking, social engineering)
- (3) Content farm—"large amounts of textual content which is specifically designed to satisfy algorithms for maximal retrieval by automated search engines." (Related: click farm—"where a large group of low-paid workers are hired to click on paid advertising links for the click fraudster.")
- (3) Micropayment—"A financial transaction involving a very small sum of money and usually one that occurs online."
- (3) Godwin's law—"If an online discussion (regardless of topic or scope) goes on long enough, sooner or later someone will compare someone or something to Hitler or Nazism." ■

Reprinted with permission of the original author. First appeared as a working document at medium.com/@yegg.

How are zlib, gzip and Zip related? What do they have in common and how are they different?

By MARK ADLER

The ZIP format supports several compression methods.

Short form

`.zip` is an archive format using, usually, the Deflate compression method. The `.gz` gzip format is for single files, also using the Deflate compression method. Often gzip is used in combination with tar to make a compressed archive format, `.tar.gz`.

The zlib library provides Deflate compression and decompression code for use by zip, gzip, png (which uses the zlib wrapper on Deflate data), and many other applications.

Long form

The ZIP format was developed by Phil Katz as an open format with an open specification, where his implementation, PKZIP, was shareware. It is an archive format that stores files and their directory structure, where each file is individually compressed. The file type is `.zip`. The files, as well as the directory structure, can optionally be encrypted.

The ZIP format supports several compression methods:

0. The file is stored (no compression)
1. The file is Shrunk
2. The file is Reduced with compression factor 1
3. The file is Reduced with compression factor 2
4. The file is Reduced with compression factor 3
5. The file is Reduced with compression factor 4
6. The file is Imploded
7. Reserved for Tokenizing compression algorithm
8. The file is Deflated
9. Enhanced Deflating using Deflate64(tm)
10. PKWARE Data Compression Library Imploding (old IBM TERSE)
11. Reserved by PKWARE
12. File is compressed using BZIP2 algorithm
13. Reserved by PKWARE
14. LZMA (EFS)
15. Reserved by PKWARE
16. Reserved by PKWARE
17. Reserved by PKWARE
18. File is compressed using IBM TERSE (new)
19. IBM LZ77 z Architecture (PFS)
97. WavPack compressed data
98. PPMd version I, Rev 1

Methods 1 to 7 are historical and are not in use. Methods 9 through 98 are relatively recent additions, and are in varying, small amounts of use. The only method in truly widespread use in the ZIP format is method 8, Deflate, and to some smaller extent method 0, which is no compression at all.

Virtually every `.zip` file that you will come across in the wild will use exclusively methods 8 and 0, and likely just method 8. (Method 8 also has the means to effectively store the data with no compression and relatively little expansion, and Method 0 cannot be streamed, whereas Method 8 can be.)

The ISO/IEC 21320-1:2015 standard for file containers is a restricted zip format, such as

Unlike .tar, .zip has a central directory at the end, which provides a list of the contents.

used in Java archive files (.jar), Office Open XML files (Microsoft Office .docx, .xlsx, .pptx), Office Document Format files (.odt, .ods, .odp), and EPUB files (.epub). That standard limits the compression methods to 0 and 8, as well as other constraints, such as no encryption or signatures.

Around 1990, the Info-ZIP group wrote portable, free, open source implementations of `zip` and `unzip` utilities, supporting compression with the Deflate format, and decompression of that and the earlier formats. This greatly expanded the use of the `.zip` format.

In the early 90's, the gzip format was developed as a replacement for the Unix `compress` utility, derived from the Deflate code in the Info-ZIP utilities.

Unix `compress` was designed to compress a single file or stream, appending a `.Z` to the file name. `compress` uses the LZW compression algorithm, which at the time was under patent and its free use was in dispute by the patent holders.

Though some specific implementations of Deflate were patented by Phil Katz, the format was not, and so it was possible to write a Deflate implementation that did not infringe on any patents. That implementation has not been so challenged in the last 20+ years.

The Unix `gzip` utility was intended as a drop-in replacement for `compress`, and in fact is able to decompress `compress`-compressed data (assuming that you were able to parse that sentence). `gzip` appends a `.gz` to the file name. `gzip` uses the Deflate compressed data format, which compresses quite a bit better than Unix `compress`, has very fast decompression, and adds a CRC-32 as an integrity check for the data.

The header format also permits the storage of more information than the `compress` format allowed, such as the original file name and the file modification time.

Though `compress` only compresses a single file, it was common to use the `tar` utility to create an archive of files, their attributes, and their directory structure into a single `.tar` file, and to then compress it with `compress` to make a `.tar.Z` file.

In fact the `tar` utility had and still has an option to do the compression at the same time, instead of having to pipe the output of `tar` to `compress`. This all carried forward to the `gzip` format, and `tar` has an option to compress directly to the `.tar.gz` format.

The `tar.gz` format compresses better than the `.zip` approach, since the compression of a `.tar` can take advantage of redundancy across files, especially many small files.

`.tar.gz` is the most common archive format in use on Unix due to its very high portability, but there are more effective compression methods in use as well, so you will often see `.tar.bz2` and `.tar.xz` archives.

Unlike `.tar`, `.zip` has a central directory at the end, which provides a list of the contents. That and the separate compression provide random access to *zlib is now in widespread use for data transmission and storage.*

the individual entries in a `.zip` file.

A `.tar` file would have to be decompressed and scanned from start to end in order to build a directory, which is how a `.tar` file is listed.

Shortly after the introduction of gzip, around the mid-1990's, the same patent dispute called into question the free use of the `.gif` image format, which was very widely used on bulletin boards and the World Wide Web (a new thing at the time).

So a small group created the PNG lossless compressed image format, with file type `.png`, to replace `.gif`. That format also uses the Deflate format for compression, which is applied after filters on the image data expose more of the redundancy.

In order to promote widespread usage of the PNG format, two free code libraries were created. libpng and zlib. libpng handled all of the features of the PNG format, and zlib provided the compression and decompression code for use by libpng, as well as for other applications. zlib was adapted from the `gzip` code.

All of the mentioned patents have since expired.

The zlib library supports Deflate compression and decompression, and three kinds of wrapping around the Deflate streams. They are: no wrapping at all ("raw" Deflate); zlib wrapping, which is used in the PNG format data blocks; and gzip wrapping, to provide gzip routines for the programmer.

The main difference between zlib and gzip wrapping is that the zlib wrapping is more compact, with six bytes vs. a minimum of 18 bytes for gzip, and the integrity check, Adler-32, runs faster than the CRC-32 that gzip uses.

Raw Deflate is used by programs that read and write the `.zip` format, which is another format that wraps around deflate compressed data.

zlib is now in widespread use for data transmission and storage. For example, most HTTP transactions by servers and browsers compress and decompress the data using zlib.

Different implementations of Deflate can result in different compressed output for the same input data, as evidenced by the existence of selectable compression levels that allow trading off compression effectiveness for CPU time.

zlib and PKZIP are not the only implementations of Deflate compression and decompression. Both the 7-Zip archiving utility and Google's zopfli library have the ability to use much more CPU time than zlib in order to squeeze out the last few bits possible when using the Deflate format, reducing compressed sizes by a few percent as compared to zlib's highest compression level.

The pigz utility, a parallel implementation of gzip, includes the option to use zlib (compression levels 1-9) or zopfli (compression level 11), and somewhat mitigates the time impact of using zopfli by splitting the compression of large files over multiple processors and cores. ■

Reprinted with permission of the original author. First appeared at stackoverflow.com/questions/.

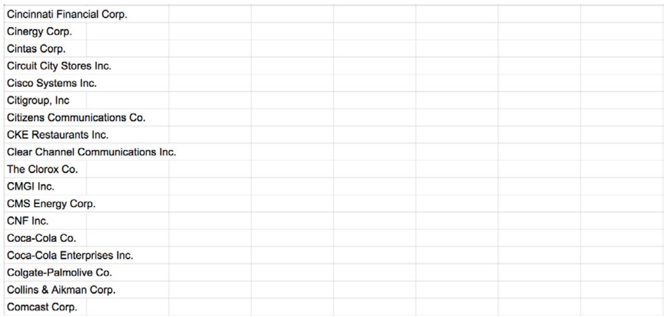
Boosting sales with machine learning

How we use natural language processing to qualify leads

By PER HARALD BORGEN

In this article, I'll explain how we're making our sales process at Xeneta more effective by training a machine learning algorithm to predict the quality of our leads based upon their company descriptions.

Head over to GitHub if you want to check out the script immediately, and feel free to suggest improvements as it's under continuous development.



Cincinnati Financial Corp.
Cinergy Corp.
Cintas Corp.
Circuit City Stores Inc.
Cisco Systems Inc.
Citigroup, Inc
Citizens Communications Co.
CKE Restaurants Inc.
Clear Channel Communications Inc.
The Clorox Co.
CMGI Inc.
CMS Energy Corp.
CNF Inc.
Coca-Cola Co.
Coca-Cola Enterprises Inc.
Colgate-Palmolive Co.
Collins & Aikman Corp.
Comcast Corp.

▲ An example of a list of potential companies to contact, pulled from sec.gov

The problem

It started with a request from business development representative Edvard, who was tired of performing the tedious task of going through big Excel sheets filled with company names, and trying to identify which ones we ought to contact.

This kind of *pre-qualification of sales leads* can take hours, as it forces the sales representative to figure out what every single company does (e.g. through reading about them on LinkedIn) so that he/she can do a qualified guess at whether or not the company is a good fit for our SaaS app.

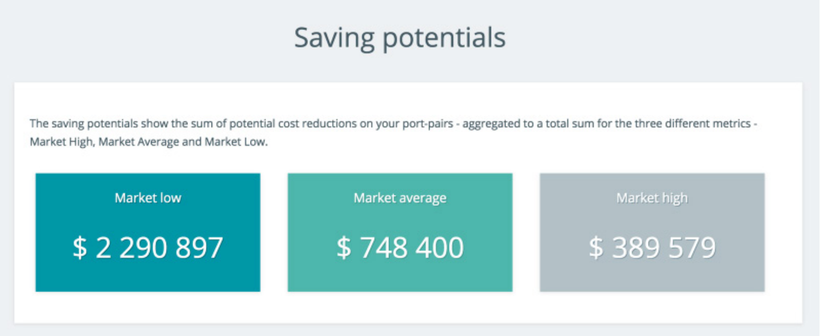
And how do you make a *qualified guess*? To understand that, you'll first need to know what we do:

> *In essence, Xeneta help companies that ship containers discover saving potential by providing sea freight market intelligence.*

More specifically, if your company ships above 500 containers per year, you're likely to discover significant saving potential by using Xeneta, as we're able to tell you exactly where you're *paying above the market average price*.

This means that our target customers are vastly different from each other, as their only common denominator is that they're somewhat involved in sea freight. Here are some examples of company categories we target:

- Automotive
- Freight forwarding
- Chemicals
- Consumer & retail
- Low paying commodities

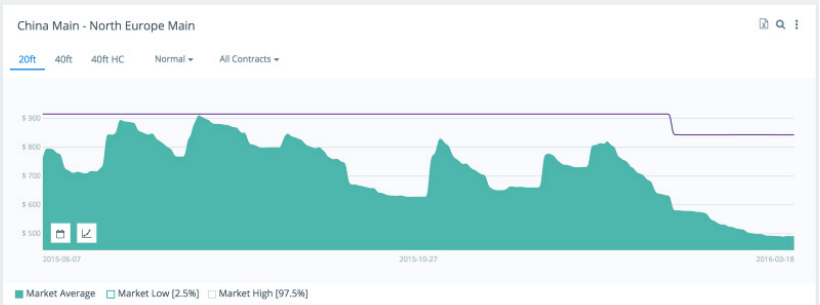


Saving potentials

The saving potentials show the sum of potential cost reductions on your port-pairs - aggregated to a total sum for the three different metrics - Market High, Market Average and Market Low.

Market low	Market average	Market high
$ 2 290 897	$ 748 400	$ 389 579

▲ This customer had a 748K USD saving potential down to market average on its sea freight spend.



▲ This widget compares a customer's contracted rate (purple line) to the market average (green graph) for 20 foot containers from China to Northern Europe.

Given a company description, can we train an algorithm to predict whether or not it's a potential Xeneta customer?

The hypothesis

Although the broad range of customers represents a challenge when finding leads, we're normally able to tell if a company is of interest for Xeneta *by reading their company description*, as it often contains hints of whether or not they're involved in sending stuff around the world.

This made us think:

> *Given a company description, can we train an algorithm to predict whether or not it's a potential Xeneta customer?*

If so, this algorithm could prove to be a huge time saver for the sales team, as it could roughly sort the Excel sheets before they start qualifying the leads manually.

The development

As I started working on this, I quickly realised that the machine learning part wasn't the only problem – we also needed a way to get hold of the company descriptions.

We considered crawling the companies' websites to fetch the *About us* section. But this smelled like a messy, unpredictable and time consuming activity, so we started looking for API's to use instead.

After some searching we discovered FullContact, which has a Company API that provides you with descriptions of millions of companies.

However, their API only accept company URLs as inputs, which are rarely present in our Excel sheets.

So we had to find a way to obtain the URLs as well, which made us land on the following workflow:

- Using the Google API to google the company name (hacky, I know…)
- Loop through the search result and find the most likely correct URL
- Use this URL to query the FullContact API

There's of course a loss at each step here, so we're going to find a better way of doing this. However, this worked well enough to test the idea out.

The dataset

Having these scripts in place, the next step was to create our training dataset. It needed to contain at least 1,000 qualified companies and 1,000 disqualified companies.

The first category was easy, as we could simply export a list of 1,000 Xeneta users from SalesForce.

Finding 1,000 disqualified was a bit tougher though, as we don't keep track of the companies we've avoided contacting. So Edvard manually disqualified 1,000 companies.

Cleaning the data

With that done, it was time to start writing the natural language processing script, with step one being to clean up the descriptions, as they are quite dirty and contain a lot of irrelevant information.

In the examples below, I'll go through each of the cleaning techniques we're currently applying, and show you how a raw description ends up as an array of numbers.

An example of a raw description:

> *Leading provider of business solutions for the global insurance industry. TIA Technology A/S is a software company that devel-*

We also need to transform the descriptions into something the machine understands, which is numbers.

ops leading edge business solutions for the global insurance industry.

RegExp

The first thing we do is to use regular expressions to get rid of non-alphabetical characters, as our model will only be able to learn words.

```
description = re.sub(“[^a-zA-Z]”, “ “, description)
```

After removing non-alphabetical characters:

Leading provider of business solutions for the global insurance industry TIA Technology A S is a software company that develops leading edge business solutions for the global insurance industry

Stemmer

We also stem the words. This means reducing multiple variations of the same word to its stem. So instead of accepting words like *manufacturer, manufaction, manufactured & manufactoring*, we rather simplify them to *manufact.*

```
from nltk.stem.snowball
import SnowballStemmer

stemmer =
SnowballStemmer(‘english’)

description =
getDescription()

description = [stemmer.
stem(word) for word in
description]
```

After stemming the words:

lead provid of busi solut for the global insur industri tia technolog a s is a softwar compani that develop lead edg busi solut for the global insur industri

Stop words

We then remove stop words, using [Natural Language Toolkit](). Stop words are words that have little relevance for the conceptual understanding of the text, such as *is, to, for, at, I, it, etc.*

```
from nltk.corpus import stopwords

stopWords = set(stopwords.
words('english'))

description =
getDescription()
```

```
description = [word for word
in description if not word in
stopWords]
```

After removing stop words:

lead provid busi solut global insur industri tia technolog softwar compani develop lead edg busi solut global insur industri

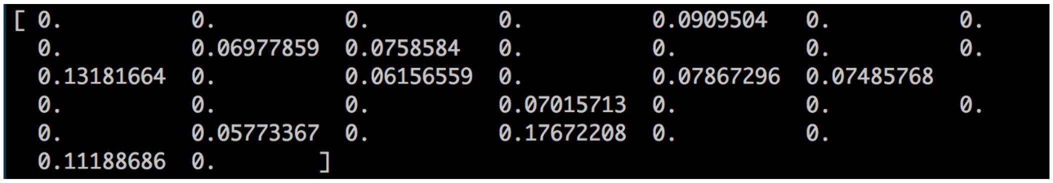
Transforming the data

But cleaning and stemming the data won’t actually help us do any machine learning, as we also need to transform the descriptions into something the machine understands, which is numbers.

Bag of Words

For this, we’re using the Bag of Words (BoW) approach. If you’re not familiar with BoW, I’d recommend you read [this Kaggle tutorial]().

BoW is a simple technique to turn *text phrases into vectors*, where each item in the vectors represents a specific word. [Scikit learn’s CountVectorizer]() gives you a super simple way to do this:



```
[ 0.          0.          0.          0.          0.0909504   0.          0.
  0.          0.06977859  0.0758584   0.          0.          0.          0.
  0.13181664  0.          0.06156559  0.          0.07867296  0.07485768
  0.          0.          0.          0.07015713  0.          0.          0.
  0.          0.05773367  0.          0.17672208  0.          0.
  0.11188686  0.        ]
```

▲ Figure 1: he vector after applying tf-idf (sorry about the bad formatting)

```
from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer(analyzer = ‘word’, max_features=5000)

vectorizer.fit(training_data)

vectorized_training_data = vectorizer.transform(training_data)
```

The *max_features* parameter tells the vectorizer how many words you want to have in our vocabulary. In this example, the vectorizer will include the 5000 words that occur most frequently in our dataset and reject the rest of them.

An example of a very small (35 items) Bag of Words vector (Ours is 5K items long):

```
[0 0 0 0 2 0 0 0 1 1 0 0 0 0
2 0 1 0 1 1 0 0 0 1 0 0 0 0 1
0 2 0 0 2 0]
```

Tf-idf transformation

Finally, we also apply a *tf-idf* transformation, which is short for *term frequency inverse document frequency*. It’s a technique that adjusts the importance of the different words in your documents.

More specifically, tf-idf will emphasise words that occur frequently in a description (*term frequency*), while de-emphasising words that occur frequently in the entire dataset (*inverse document frequency*).

```
from sklearn.feature_extraction.text import TfidfTransformer
tfidf =
TfidfTransformer(norm=’l1')

tfidf.fit(vectorized_training_data)

tfidf_vectorized_data = tfidf.transform(vectorized_training_data)
```

Again, scikit learn saves the day by providing tf-idf out of the box. Simply fit the model to your vectorized training data, and then use the transform method to transform it. (See figure 1)

The algorithm

After all the data has been *cleaned, vectorised and transformed*, we can finally start doing some machine learning, which is one of the simplest parts of this task.

I first sliced the data into 70% training data and 30% testing data, and then started off with two scikit learn algorithms: Random Forest (RF) and K Nearest Neighbors (KNN).

It quickly became clear that RF outperformed KNN, as the former quickly reached more than 80% accuracy while the latter stayed at 60%.

Fitting a scikit learn model is

```
def runForest(X_train, X_test, Y_train, Y_test):

       forest = RandomForestClassifier(n_estimators=100)
       forest = forest.fit(X_train, Y_train)
       score = forest.score(X_test, Y_test)
       return score

forest_score = runForest(X_train, X_test, Y_train, Y_test)
```

▲ Figure 2: Finding scikit learn model

super simple. (See figure 2)

So I continued with RF to see how much I could increase the accuracy by tuning the following parameters:

- *Vocabulary:* how many words the CountVectorizer includes in the vocabulary (currently 5K)
- *Gram Range:* size of phrases to include in Bag Of Words (currently 1–3, meaning up until '3 word'-phrases)
- *Estimators:* amount of estimators to include in Random Forest (currently 90)

With these parameters tuned, the algorithm reaches an accuracy of 86.4% on the testing dataset, and is actually starting to become useful for our sales team.

The road ahead

However, the script is by no means finished. There are *tons* of ways to improve it. For example, the algorithm is likely to be biased towards the kind of descriptions we currently have in our training data. This might become a performance bottleneck when testing it on more real world data.

Here are a few activities we're considering to do in the road ahead:

- Get more data (scraping, other API's, improve data cleaning)
- Test other types of data transformation (e.g. word2vec)
- Test other ML algorithms (e.g. neural nets)

We'll be pushing to GitHub regularly if you want to follow the progress. And feel free to leave a comment below if you have anything you'd like to add. ■

Thanks for reading! We are Xeneta — the world's leading sea freight intelligence platform. We're always looking for bright minds to join us, so head over to our website if you're interested! You can follow us at both Twitter and Medium.

Reprinted with permission of the original author. First appeared at medium.com/xeneta/.

We built voice modulation to mask gender in technical interviews. Here’s what happened.

By ALINE LERNER

We made men sound like women and women sound like men, and looked at how that affected their interview performance.

Interviewing.io is a platform where people can practice technical interviewing anonymously and in the process, find jobs based on their interview performance rather than their resumes.

Since we started, we've amassed data from thousands of technical interviews, and in this blog, we routinely share some of the surprising stuff we've learned. In this post, I'll talk about what happened when we built real-time voice masking to investigate the magnitude of bias against women in technical interviews.

In short, we made men sound like women and women sound like men, and looked at how that affected their interview performance. We also looked at what happened when women did poorly in interviews, how drastically that differed from men's behavior, and why that difference matters for the thorny issue of the gender gap in tech.

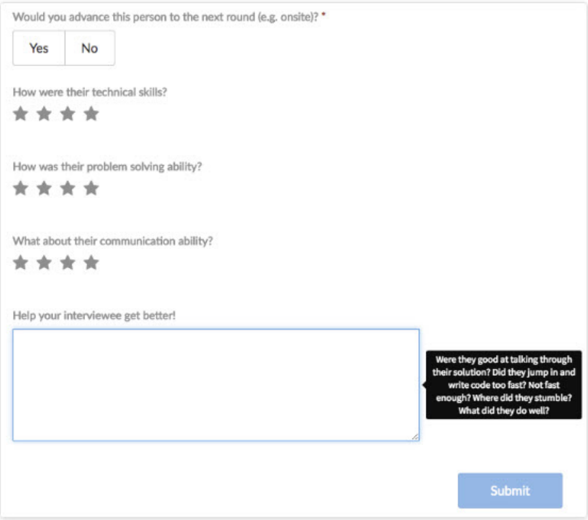
The setup

When an interviewer and an interviewee match on our platform, they meet in a collaborative coding environment with voice, text chat, and a whiteboard and jump right into a technical question.

Interview questions on the platform tend to fall into the category of what you'd encounter at a phone screen for a back-end software engineering role, and interviewers typically come from a mix of large companies like Google, Facebook, Twitch, and Yelp, as well as engineering-focused startups like Asana, Mattermark, and others.

After every interview, interviewers rate interviewees on a few different dimensions.

As you can see, we ask the interviewer if they would advance their interviewee to the next round. We also ask about a few different aspects of interview performance using a 1-4 scale. On our platform, a score of 3 or above is generally considered good.



Would you advance this person to the next round (e.g. onsite)? *

How were their technical skills?

★ ★ ★ ★

How was their problem solving ability?

★ ★ ★ ★

What about their communication ability?

★ ★ ★ ★

Help your interviewee get better!

Were they good at talking through their solution? Did they jump in and write code too fast? Not fast enough? Where did they stumble? What did they do well?

▲ Feedback form for interviewers

Women historically haven't performed as well as men…

One of the big motivators to think about voice masking was the increasingly uncomfortable disparity in interview performance on the platform between

Armed with the ability to hide gender during technical interviews, we were eager to see what the hell was going on...

men and women[1]. At that time, we had amassed over a thousand interviews with enough data to do some comparisons and were surprised to discover that women really were doing worse.

Specifically, *men were getting advanced to the next round 1.4 times more often than women. Interviewee technical score wasn't faring that well either — men on the platform had an average technical score of 3 out of 4, as compared to a 2.5 out of 4 for women.*

Despite these numbers, it was really difficult for me to believe that women were just somehow worse at computers, so when some of our customers asked us to build voice masking to see if that would make a difference in the conversion rates of female candidates, we didn't need much convincing.

... so we built voice masking

Since we started working on interviewing.io, in order to achieve true interviewee anonymity, we knew that hiding gender would be something we'd have to deal with eventually but we put it off for a while because it wasn't technically trivial to build a real-time voice modulator. Some early ideas included sending female users a Bane mask.

When the Bane mask thing didn't work out, we decided we ought to build something within the app, and if you play the videos below, you can get an idea of what voice masking on interviewing.io sounds like. In the first one, I'm talking in my normal voice.

And in the second one, I'm modulated to sound like a man.[2]

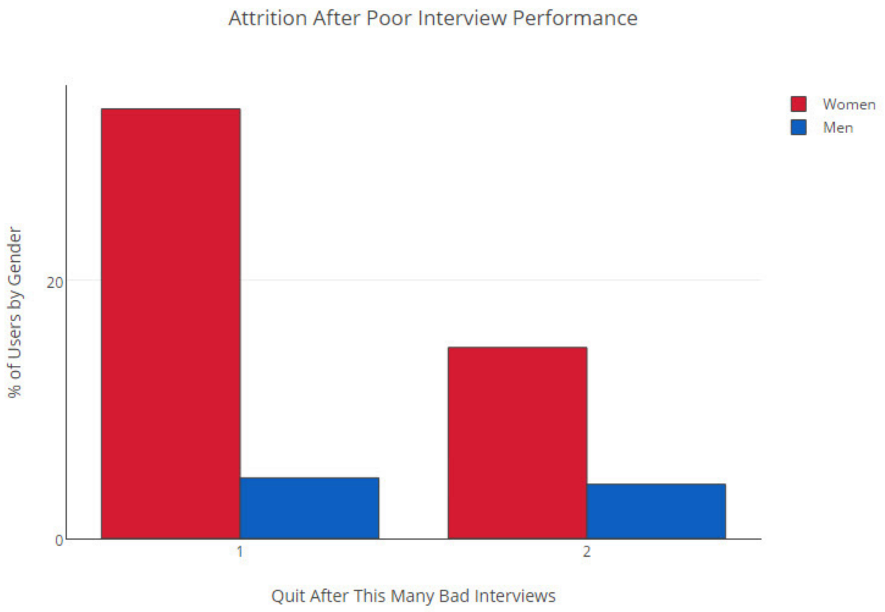
Armed with the ability to hide gender during technical interviews, we were eager to see what the hell was going on and get some insight into why women were consistently underperforming.

The experiment

The setup for our experiment was simple. Every Tuesday evening at 7 PM Pacific, interviewing.io hosts what we call practice rounds. In these practice rounds, anyone with an account can show up, get matched with an interviewer, and go to town.

And during a few of these rounds, *we decided to see what would happen to interviewees' performance when we started messing with their perceived genders.*

In the spirit of not giving away what we were doing and potentially compromising the experiment, we told both interviewees and interviewers that we were slowly rolling out our new voice masking feature and that



they could opt in or out of helping us test it out. Most people opted in, and we informed interviewees that their voice might be masked during a given round and asked them to refrain from sharing their gender with their interviewers. For interviewers, we simply told them that interviewee voices might sound a bit processed.

We ended up with 234 total interviews (roughly 2/3 male and 1/3 female interviewees), which fell into one of three categories:

- Completely unmodulated (useful as a baseline)
- Modulated without pitch change
- Modulated with pitch change

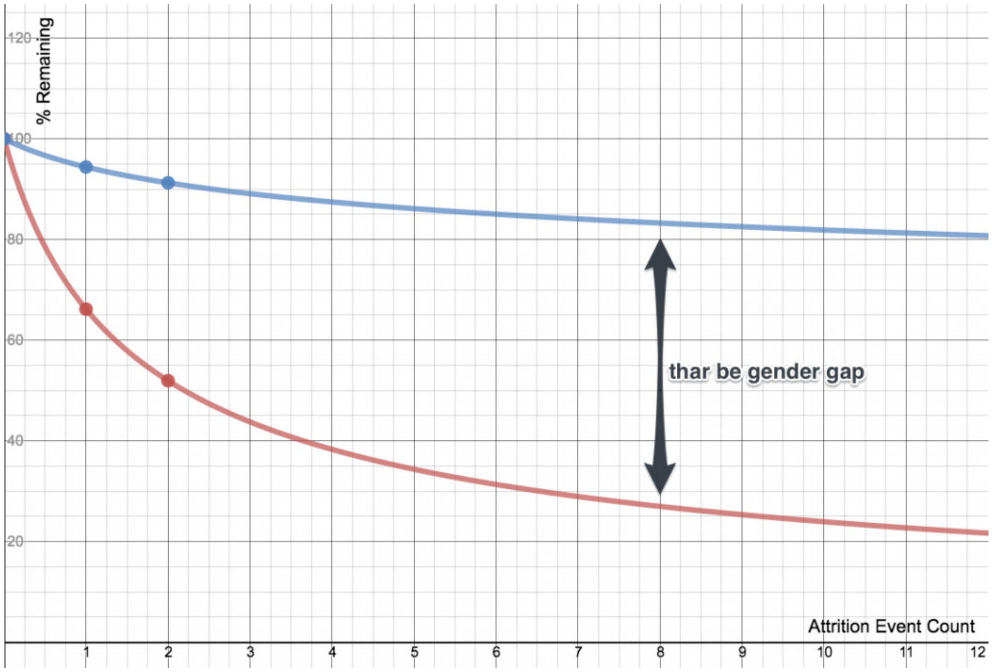
You might ask why we included the second condition, i.e. modulated interviews that didn't change the interviewee's pitch. As you probably noticed, if you played the videos above, the modulated one sounds fairly processed.

The last thing we wanted was for interviewers to assume that any processed-sounding interviewee must summarily have been the opposite gender of what they sounded like. So we threw that condition in as a further control.

The results

After running the experiment, we ended up with some rather surprising results. *Contrary to what we expected* (and probably contrary to what you expected as well!), *masking gender had no effect on interview performance* with respect to any of the scoring criteria (would advance to next round, technical ability, problem solving ability).

If anything, we started to notice some trends in the opposite direction of what we expected:



for technical ability, it appeared that men who were modulated to sound like women did a bit better than unmodulated men, and that women who were modulated to sound like men did a bit worse than unmodulated women. Though these trends weren't statistically significant, I am mentioning them because they were unexpected and definitely something to watch for as we collect more data.

On the subject of sample size, we have no delusions that this is the be-all and end-all of pronouncements on the subject of gender and interview performance. We'll continue to monitor the data as we collect more of it, and it's very possible that as we do, everything we've found will be overturned.

I will say, though, that had there been any staggering gender bias on the platform, with a few hundred data points, we would have gotten some kind of result. So that, at least, was encouraging.

So if there's no systemic bias, why are women performing worse?

After the experiment was over, I was left scratching my head. If the issue wasn't interviewer bias, what could it be? I went back and looked at the seniority levels of men vs. women on the platform as well as the kind of work they were doing in their current jobs, and neither of those factors seemed to differ significantly between groups.

Women leave interviewing.io roughly 7 times as often as men after they do badly in an interview.

But there was one nagging thing in the back of my mind. I spend a lot of my time poring over interview data, and I had noticed something peculiar when observing the behavior of female interviewees. Anecdotally, it seemed like women were leaving the platform a lot more often than men. So I ran the numbers.

What I learned was pretty shocking. *As it happens, women leave interviewing.io roughly 7 times as often as men after they do badly in an interview.* And the numbers for two bad interviews aren't much better.

You can see the breakdown of attrition by gender below (the differences between men and women are indeed statistically significant with P < 0.00001).

Also note that as much as possible, I corrected for people leaving the platform because they found a job (practicing interviewing isn't that fun after all, so you're probably only going to do it if you're still looking), were just trying out the platform out of curiosity, or they didn't like something else about their interviewing.io experience.

A totally speculative thought experiment

So, if these are the kinds of behaviors that happen in the interviewing.io microcosm, how much is applicable to the broader world of software engineering? Please bear with me as I wax hypothetical and try to extrapolate what we've seen here to our industry at large. And also, please know that what follows is very speculative, based on not that much data, and could be totally wrong… but you gotta start somewhere.

If you consider the attrition data points above, you might want to do what any reasonable person would do in the face of an existential or moral quandary, i.e. fit the data to a curve. An exponential decay curve seemed reasonable for attrition behavior, and you can see what I came up with below.

The x-axis is the number of what I like to call "attrition events", namely things that might happen to you over the course of your computer science studies and subsequent career that might make you want to quit.

The y-axis is what portion of people are left after each attrition event. The red curve denotes women, and the blue curve denotes men.

Now, as I said, this is pretty speculative, but it really got me thinking about what these curves might mean in the broader context of women in computer science. How many "attrition events" does one encounter between primary and secondary

We need 3 times as many women studying computer science than men to get to the same number in our pipelines.

education, and entering a collegiate program in CS and then starting to embark on a career?

So, I don't know, let's say there are 8 of these events between getting into programming and looking around for a job. If that's true, then we need 3 times as many women studying computer science than men to get to the same number in our pipelines.

Note that that's 3 times more than men, not 3 times more than there are now. If we think about how many there are now, which, depending on your source, is between 1/3 and a 1/4 of the number of men, *to get to pipeline parity, we actually have to increase the number of women studying computer science by an entire order of magnitude.*

Prior art, or why maybe this isn't so nuts after all

Since gathering these findings and starting to talk about them a bit in the community, I began to realize that there was some supremely interesting academic work being done on gender differences around self-perception, confidence, and performance.

Some of the work below found slightly different trends than we did, but it's clear that anyone attempting to answer the question of the gender gap in tech would be remiss in not considering the effects of confidence and self-perception in addition to the more salient matter of bias.

In a study investigating the effects of perceived performance to likelihood of subsequent engagement, Dunning (of Dunning-Kruger fame) and Ehrlinger administered a scientific reasoning test to male and female undergrads, and then asked them how they did.

Not surprisingly, though there was no difference in performance between genders, women underrated their own performance more often than men. Afterwards, participants were asked whether they'd like to enter a Science Jeopardy contest on campus in which they could win cash prizes. Again, women were significantly less likely to participate, with participation likelihood being directly correlated with self-perception rather than actual performance.[3]

In a different study, sociologists followed a number of male and female STEM students over the course of their college careers via diary entries authored by the students. One prevailing trend that emerged immediately was the difference between how men and women handled the "discovery of their [place in the]

It's about women being bad at dusting themselves off after failing, which, despite everything, is probably a lot easier to fix.

pecking order of talent, an initiation that is typical of socialization across the professions."

For women, realizing that they may no longer be at the top of the class and that there were others who were performing better, "the experience [triggered] a more fundamental doubt about their abilities to master the technical constructs of engineering expertise [than men]."

And of course, what survey of gender difference research would be complete without an allusion to the wretched annals of dating? When I told the interviewing.io team about the disparity in attrition between genders, the resounding response was along the lines of, "Well, yeah. Just think about dating from a man's perspective."

Indeed, a study published in the *Archives of Sexual Behavior* confirms that men treat rejection in dating very differently than women, even going so far as to say that men "reported they would experience a more positive than negative affective response after… being sexually rejected."

Maybe tying coding to sex is a bit tenuous, but, as they say, programming is like sex — one mistake and you have to support it for the rest of your life.

Why I'm not depressed by our results and why you shouldn't be either

Prior art aside, I would like to leave off on a high note. I mentioned earlier that men are doing a lot better on the platform than women, but here's the startling thing. *Once you factor out interview data from both men and women who quit after one or two bad interviews, the disparity goes away entirely.*

So while the attrition numbers aren't great, I'm massively encouraged by the fact that at least in these findings, it's not about systemic bias against women or women being bad at computers or whatever. Rather, it's about women being bad at dusting themselves off after failing, which, despite everything, is probably a lot easier to fix. ■

[1] Roughly 15% of our users are female. We want way more, but it's a start.

[2] If you want to hear more examples of voice modulation or are just generously down to indulge me in some shameless bragging, we got to demo it on NPR and in Fast Company.

[3] In addition to asking interviewers how interviewees did, we also asked interviewees to rate themselves. After reading the Dunning and Ehrlinger study, we went back and checked to see what role self-perception played in attrition. In our case, the answer is, I'm afraid, TBD, as we're going to need more self-ratings to say anything conclusive.

Reprinted with permission of the original author. First appeared at blog.interviewing.io.

A simple mind hack that helps beat procrastination

By SHYAL BEARDSLEY

Imagine yourself starting, not finishing.

Cleaning the flat, working out, reaching inbox zero: these are all worthy goals, yet sometimes getting stuck in the infernal hackersnews ∞ reddit loop seems so much more... relaxing.

Let's begin by delving into the physiology of procrastination:

> *"[procrastination is] a battle of the limbic system, the unconscious zone that includes the pleasure center, and the prefrontal cortex, the internal planner. When the limbic system dominates, which is pretty often, the result is putting off until tomorrow what could, and should, be done today."*
>
> *Source:* scienceabc.com

So how can we trick our limbic system to not "dominate"? The answer resides in how we picture the task. When we picture a task, we often imagine completing the task, if such a thing is even possible, and are also aware of task inter-relatedness. Therein lies the problem.

This is what Neil Fiore has to say about it in his book: The Now Habit.

> *"The task before you is to walk a solid board that is thirty feet long, four inches thick, and one foot wide. You have all the physical, mental, and emotional abilities necessary to perform this task. You can carefully place one foot in front of the other, or you can dance, skip, or leap across the board. You can do it. No problem."*

This is how non-procrastinators visualise a task. However, Situation B is how procrastinators visualise the same task.

> *"Now imagine that the task is just the same, to walk a board thirty feet long and one foot wide, and you have the same abilities; only now the board is suspended between two buildings 100 feet above the pavement. Look across to the other end of the board and contemplate beginning your assignment. What do you feel? What are you thinking about? What are you saying to yourself?"*

The task is more or less the same; however the mental representation is entirely different.

Imagine yourself *starting*, not finishing

What is the first, smallest, shortest, and least effortful task you can perform to get started? Say you need to clean your flat: then it could be to:

> *take 1 plate and put it in the sink.*

That simple act has now set you in motion. In other words: *the prefrontal cortex has won*.

Why does this work?

It works because your limbic system is now experiencing the previously terrifying task, but is no longer feeling terrified by it. This enables you to build new, fresh memories of what was previously uncomfortable to the point of procrastination, but now feels perfectly fine and safe.

tldr: imagine yourself starting, not finishing. ■

Reprinted with permission of the original author. First appeared at shyal.com/blog.

food bit *

The unmanly fork

These days, eating with a fork can't be any more unremarkable, but way before this table tool gained acceptance, it was viewed as unmanly and ridiculous. After all, most people ate with their fingers and anything that can't be picked up were scooped up by spoons or cut up by knives.

The early adopters of the fork were the 11th century denizens of the Byzantine Empire, who astonished foreigners by delicately spearing their food with forks. This practice, however, didn't take off until 600 years later when the Europeans embraced the fork and established rules for its usage. The Americans, late to the party, only took up forks during the late 18th century, and by then, the explosion of specialized forks (fish fork, anyone?) had some wishing they'd never picked up forks in the first place.

* FOOD BIT is where we, enthusiasts of all edibles, sneak in a fun fact about food.

HACKER BITS is the monthly magazine that gives you the hottest technology stories crowdsourced by the readers of Hacker News. We select from the top voted stories and publish them in an easy-to-read magazine format.

Get HACKER BITS delivered to your inbox every month! For more, visit hackerbits.com.