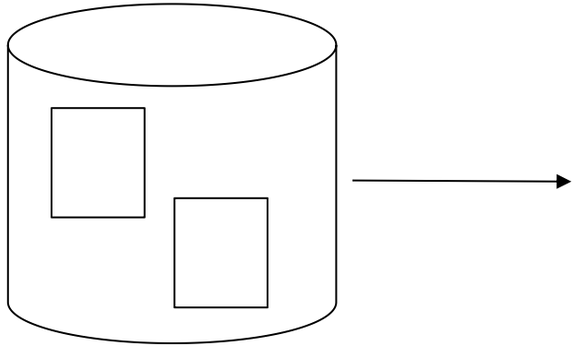
Data Extraction

AnHai Doan

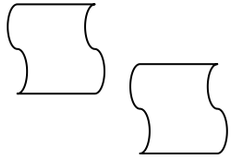
Motivation



X

id	name	loc
x_1	Apple	CA
x_2	IBM	NY

news articles



data extraction

Y

id	cname	address	rev
y_1	IBM Corp	CA	25
y_2	Apple Inc	CA	51
y_3	GE	NY	351

X

id	name	loc
x_1	Apple	CA
x_2	IBM	NY

Y

id	cname	address	rev
y_1	IBM Corp	CA	25
y_2	Apple Inc	CA	51
y_3	GE	NY	351

→

Z

name	loc	rev
Apple Inc	CA	51
IBM Corp	NY	25

Overview

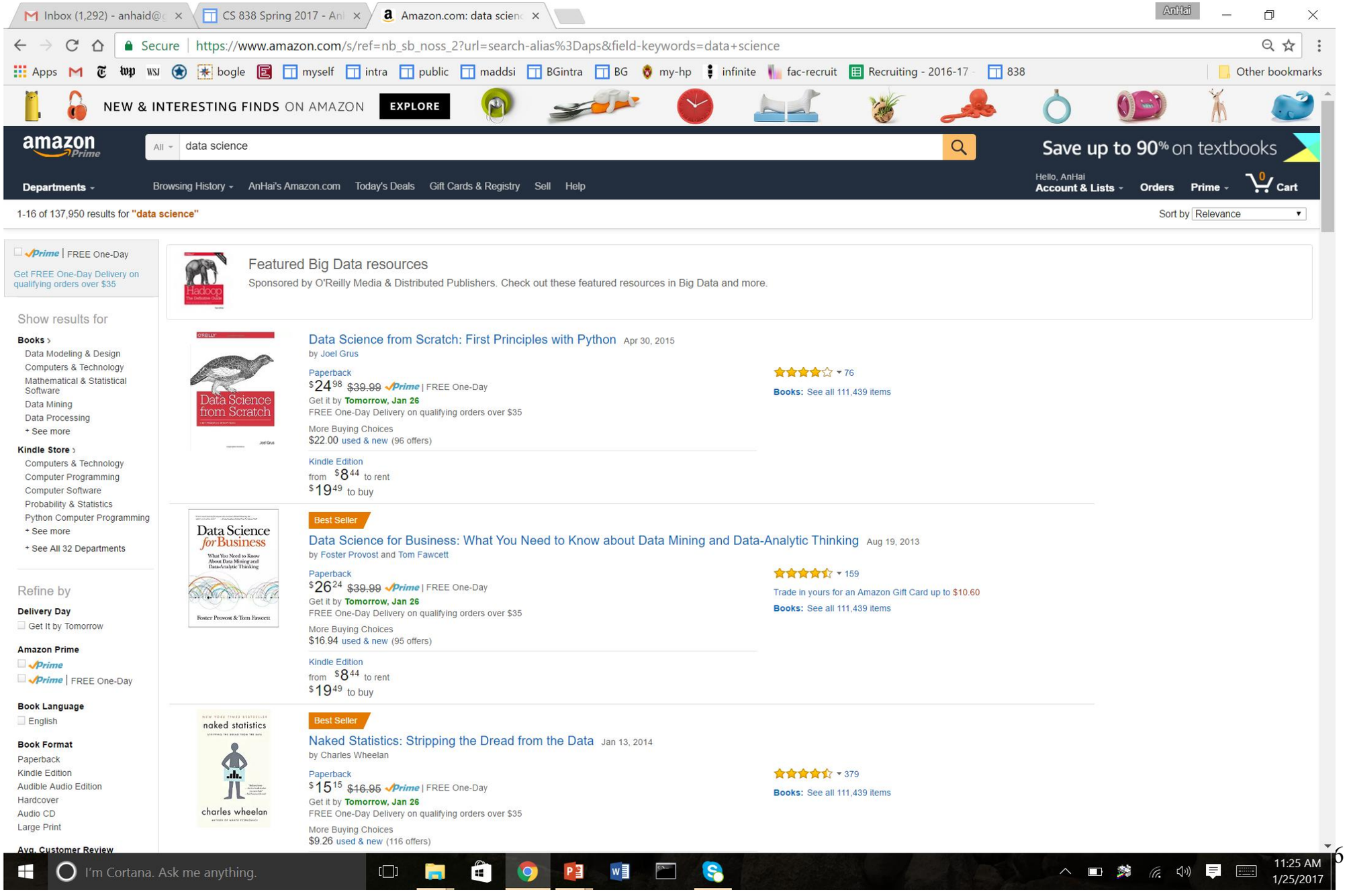
- **Two types of extraction**
  - from template-based pages
  - from text
- **Two types of methods**
  - rule-based
  - ML

Extracting from Template-Based Pages

- **See the example in next slide**

- **Why do we have template-based data?**
  - An example on how this data is generated
  - Querying on Amazon by filling in a form interface using Data Science
  - The query goes to a database in the backend
  - Database result is plugged into template-based pages
  - These pages are presented to the user

Template-Based Pages



Template = Schema + Format

- **Schema**
  - what attributes, in what order
- **Format**
  - HTML formatting

(a) *countries.com* (b) *easycalls.com* (c) *greatbooks.com*

Figure 11.1: Examples of data sources, each of which displays data using a schema and a format that are common across all pages of the source.

The Extraction Problem

- We assume a Web site with many pages (say hundreds or thousands or more)
- All pages conform to the same schema plus display format
- Our goal is to extract the data that conform to the schema from all pages

Manually Writing Extraction Rules

(a)

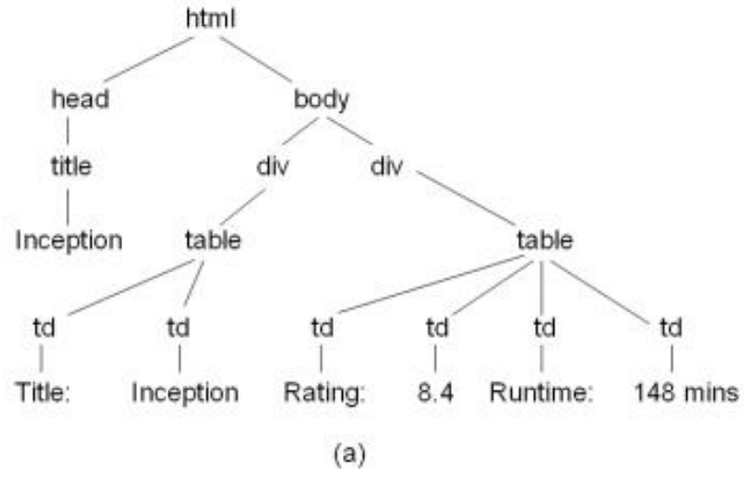
```
#!/usr/bin/perl -w

open(INFILE, $ARGV[0]) or die "can't open file\n";
while ($line = <INFILE>) {
  if ($line =~ m/<B\>(.+?)\<\/B\>\s+?\<I\>(\d+?)\<\/I\>\<BR\>/) {
      print "($1,$2)\n";
  }
}
close(INFILE);
```

(b)

Figure 11.2: (a) An example page from source *easycalls.com*, and (b) a manually constructed wrapper (a Perl program in this case) that extracts data from such pages.

Manually Writing Extraction Rules



title = */html/body/div[1]/table/td[2]/text()*
rating = */html/body/div[2]/table/td[2]/text()*
runtime = */html/body/div[2]/table/td[4]/text()*

(b)

Figure 11.3: (a) The DOM tree of a Web page about a movie, and (b) a wrapper that uses the DOM tree to extract the title, rating, and run time of the movie.

Manually Writing Extraction Rules

- **We would use a set of pages to write the extraction rules**
  - these pages are called the development set
- **Then apply the rules to a new set of pages**
  - then check accuracy and refine the rules
- **And so on …**
- **When we think the rules are ready, we apply them to all the remaining pages from the Web site**

Using ML


```
<HTML>
<TITLE>Countries in Australia (Continent)</TITLE>
<BODY>
<B>Countries in Australia (Continent)</B><P>
<B>Australia</B>  <I>61</I><BR>
<B>East Timor</B>  <I>670</I><BR>
<B>Papua New Guinea</B>  <I>675</I><BR>
<HR>
<B>Copyright easycalls.com</B>
</BODY>
</HTML>
```

head

data region

tail

- **We want to extract pairs of <country, calling code>**
- **Ideally, we want to use ML to learn an extraction program, which is a piece of code**
- **But this would be way too hard**
- **So we will make assumptions to simplify what we need to learn**
  - specifically, we will only have to learn six strings
  - if we know them, we can automatically write the extraction program

Using ML


```
<HTML>
<TITLE>Countries in Australia (Continent)</TITLE>
<BODY>
<B>Countries in Australia (Continent)</B><P>
<B>Australia</B>  <I>61</I><BR>
<B>East Timor</B>  <I>670</I><BR>
<B>Papua New Guinea</B>  <I>675</I><BR>
<HR>
<B>Copyright easycalls.com</B>
</BODY>
</HTML>
```

head

data region

tail

- **Assumptions**
  - page starts with a header which ends with a string H
  - page ends with a tail which starts with a string T
  - in between, each country starts with string A and ends with string B
  - each calling code starts with string C and ends with string D
- **So the ML model is <H, T, A, B, C, D>**
  - we use the training data to find these

Using ML

- Let the set of all pages on the Web site be D
- We take a small set S of D to be the training set
- We ask the user to label all countries and calling codes in S
- Then we use S to learn the six strings
- If we need more pages for training, then we can take more pages from D and ask user to label those
- Once we think we have learned the six strings, we can write the extraction program and apply it to all remaining pages in D to extract countries and calling codes

Extracting Entities/Attributes/Relationships from Text, called “IE from Text”

Extract Entities, Attributes, Relations

For years, Microsoft Corporation CEO Bill Gates was against open source. But today he appears to have changed his mind. "We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

```
Select  Name
From    PEOPLE
Where Organization = 'Microsoft'
```

PEOPLE

Name	Title	Organization
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	Founder	Free Soft..

Bill Gates

Bill Veghte

(from Cohen's IE tutorial, 2003)

Extract Conference Names

Extract Entities and Attributes from Products

Attribute	Walmart Product	Vendor Product
Product Name	CHAMP Bluetooth Survival Solar Multi-Function Skybox with Emergency AM/FM NOAA Weather Radio (RCEP600WR)	CHAMP Bluetooth Survival Solar Multi-Function Skybox with Emergency AM/FM NOAA Weather Radio (RCEP600WR)
Product Short Description	BLTH SURVIVAL SKYBOX W WR	
Product Long Description	BLTH SURVIVAL SKYBOX W WR	BLTH SURVIVAL SKYBOX W WR
Product Segment	Electronics	Electronics
Product Type	CB Radios & Scanners	Portable Radios
Color	Black	
Actual Color	Black	
UPC		0004447611732

Attribute	Walmart Product	Vendor Product
Product Name	GreatShield 6FT Apple MFi Licensed Lightning Sync Charge Cable for Apple iPhone 6 6 Plus 5S 5C 5 iPad 4 Air Mini - Black	GreatShield 6FT Apple MFi Licensed Lightning Sync Charge Cable for Apple iPhone 6 6 Plus 5S 5C 5 iPad 4 Air Mini - White
Product Short Description	GreatShield 6FT Apple MFi Licensed Lightning Sync Charge Cable for Apple iPhone 6 6 Plus 5S 5C 5 iPad 4 Air Mini - Black	
Product Long Description	**GreatShield Apple MFi Licensed Lightning Charge & Sync Cable** This USB 2.0 cable connects your iPhone, iPad, or iPod with Lightning …	**GreatShield Apple MFi Licensed Lightning Charge & Sync Cable** This USB 2.0 cable connects your iPhone, iPad, or iPod with Lightning …
Product Segment	Electronics	Electronics
Product Type	Cable Connectors	Cable Connectors
Brand	GreatShield	GreatShield
Manufacturer Part Number	GS09055	

Sometimes We Need to Do Both

- **Wrapper-based extraction first, then IE**
  - e.g., extract products from Amazon pages, then IE from text on products

Two Main Solution Approaches

- Hand-crafted rules
  - Eg regexes
  - Dictionary based
- Learning-based approaches

Example: Regexes

- **Extract attribute values from products**

title	= X-Mark Pair of 45 lb. Rubber Hex Dumbbells
material	= Rubber
finer categorizations	= Dumbbells__Weight Sets
type	= Hand Weights
…	

title	= Zalman ZM-T2 ATX Mini Tower Case - Black
brand	= Zalman
finer categorizations	= Computer Cases
…	

- **Discuss how to extract weights such as 45 lbs**
  - Something to recognize the number
  - Something to recognize all variations of weight units
  - The resulting regex can be very complicated

Example: Dictionary Based

- Goal: build a simple person-name extractor
  - input: a set of Web pages W, a list of names
  - output: all mentions of names in W

- Simplified Person-Name extraction
  - for each name e.g., David Smith
    - generate variants (V): “David Smith”, “D. Smith”, “Smith, D.”, etc.
    - find occurrences of these variants in W
  - clean the occurrences

Compiled Dictionary

…….
…….
…….
…….
…….
…….
…….
David Miller
Rob Smith
Renee Miller

D. Miller, R. Smith, K. Richard, D. Li

Hand-coded rules can be arbitrarily complex

Find conference name in raw text

```
##########################################################################
# Regular expressions to construct the pattern to extract conference names
##########################################################################

# These are subordinate patterns
my $wordOrdinals="(?:first|second|third|fourth|fifth|sixth|seventh|eighth|ninth|tenth|eleventh|twelfth|thirteenth|fourteenth|fifteenth)";
my $numberOrdinals="(?:\\d?(?:1st|2nd|3rd|1th|2th|3th|4th|5th|6th|7th|8th|9th|0th))";
my $ordinals="(?:$wordOrdinals|$numberOrdinals)";
my $confTypes="(?:Conference|Workshop|Symposium)";
my $words="(?:[A-Z]\\w+\\s*)"; # A word starting with a capital letter and ending with 0 or more spaces
my $confDescriptors="(?:international\\s+|[A-Z]+\\s+)"; # .e.g "International Conference ...' or the conference name for workshops (e.g.
"VLDB Workshop ...")
my $connectors="(?:on|of)";
my $abbreviations="(?:\\([A-Z]\\w\\w+[\\W\\s]*?(?:\\d\\d+)?\\))"; # Conference abbreviations like "(SIGMOD'06)"

# The actual pattern we search for.  A typical conference name this pattern will find is
# "3rd International Conference on Blah Blah Blah (ICBBB-05)"
my
$fullNamePattern="((?:$ordinals\\s+$words*|$confDescriptors)?$confTypes(?:\\s+$connectors\\s+.*?|\\s+)?$abbreviations?)(?:\\n|\\r|\\.|<)";

############################## ################################
# Given a <dbworldMessage>, look for the conference pattern
#############################################################
lookForPattern($dbworldMessage, $fullNamePattern);

#########################################################
# In a given <file>, look for occurrences of <pattern>
# <pattern> is a regular expression
#########################################################
sub lookForPattern {
my ($file,$pattern) = @_;
```

Example Code of Hand-Coded Extractor

```
# Only look for conference names in the top 20 lines of the file
my $maxLines=20;
my $topOfFile=getTopOfFile($file,$maxLines);

# Look for the match in the top 20 lines - case insenstive, allow matches spanning multiple lines
if($topOfFile=~/(.*?)$pattern/is) {
my ($prefix,$name)=($1,$2);

# If it matches, do a sanity check and clean up the match
# Get the first letter
# Verify that the first letter is a capital letter or number
if(!($name=~/^\W*?[A-Z0-9]/)) { return (); }

# If there is an abbreviation, cut off whatever comes after that
if($name=~/^(.*?$abbreviations)/s) { $name=$1; }

# If the name is too long, it probably isn't a conference
if(scalar($name=~/[^\s]/g) > 100) { return (); }

# Get the first letter of the last word (need to this after chopping off parts of it due to abbreviation
my ($letter,$nonLetter)=("[A-Za-z]","[^A-Za-z]");
" $name"=~/$nonLetter($letter) $letter*$nonLetter*$/; # Need a space before $name to handle the first $nonLetter in the pattern if there
is only one word in name

my $lastLetter=$1;
if(!($lastLetter=~/[A-Z]/)) { return (); } # Verify that the first letter of the last word is a capital letter

# Passed test, return a new crutch
return newCrutch(length($prefix),length($prefix)+length($name),$name,"Matched pattern in top $maxLines lines","conference
name",getYear($name));
}
return ();
}
```

The Learning Based Approach

- What is the overall process
- Starting with a set of pages D

Two Main Solution Approaches

- Hand-crafted rules
  - Eg regexes
  - Dictionary based
- Learning-based approaches
  - we discussed this earlier
  - discuss how to compute accuracy: precision, recall, F1
  - discuss computing different types of accuracy
- Pros and cons of the two approaches

Other Kinds of Data Extraction

- Excel
- PDF
- Images
- Extracting a table from textual document, from HTML
- Discover and extract structures from log files, CVS files, etc.

Extraction using LLMs

Inbox (3,651) - anhaid@gmail.com

ChatGPT

chat.openai.com/c/e4b7af07-3593-4810-90d1-40c5c80f1a18

my-hp my-uw-zoom my-drive my-cal weekly-schedule Canvas All Bookmarks

ChatGPT 4

You

Extract person names from the following text "For years, Microsoft Corporation CEO Bill Gates was against open source. But today he appears to have changed his mind. "We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying...
"

ChatGPT

The person names extracted from the provided text are:

- Bill Gates
- Bill Veghte
- Richard Stallman

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Inbox (3,651) - anhaid@gmail.com

ChatGPT

chat.openai.com/c/e4b7af07-3593-4810-90d1-40c5c80f1a18

my-hp my-uw-zoom my-drive my-cal weekly-schedule Canvas All Bookmarks

ChatGPT 4

- Bill Veghte
- Richard Stallman

You

extract all tuples of the form (person name, title, organization) from the above text

ChatGPT

The tuples of the form (person name, title, organization) extracted from the provided text are:

- (Bill Gates, CEO, Microsoft Corporation)
- (Bill Veghte, VP, Microsoft)
- (Richard Stallman, founder, Free Software Foundation)

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Inbox (3,651) - anhaid@gmail.c | ChatGPT | SampleRadioCalls.pdf

chat.openai.com/c/e4b7af07-3593-4810-90d1-40c5c80f1a18

my-hp | my-uw-zoom | my-drive | my-cal | weekly-schedule | Canvas | All Bookmarks

ChatGPT 4

You

extract all plane names from the following text "Pilot: Lancaster Tower, Cherokee 8121K, Runway 26, ready
for takeoff.
Tower: Cherokee 8121K, Runway 26, cleared for takeoff.
Pilot: Cleared for takeoff Runway 26, Cherokee 8121K."

ChatGPT

The plane name extracted from the provided text is:

- Cherokee 8121K

Is this conversation helpful so far?

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.