Information Extraction for Open Data

Pablo Ariel Duboue, PhD

Data Science for Social Good Meetup
October 1st, Vancouver, BC

This talk in one slide

- Information Extraction: a series of techniques to extract unstructured information from text into structured data.
  - Very limited semantic content
- Open Data: data that can be freely used, shared and built-on by anyone, anywhere, for any purpose.
  - Governments releasing large quantities of unstructured information
- Structure
  1. AI for social good
  2. IE for OpenData project
  3. IE crash course
  4. Complexity discussion

Outline

AI is everywhere

- Artificial Intelligence through custom programs or trained models is ubiquous
- We use the models to help us decide what to buy, what to watch and who to date
- Most of these systems are built by individual actors in society with explicit goals that help themselves
  - For example, better predictive models for advertisement

The low perplexity of the unpredictable human spirit

- We are being data mined into oblivion
- The models know what we are going to do before we know that ourselves

Example: "Digital tool for landlords measures potential tenants' kindness, cleanliness"

- https://www.ctvnews.ca/business/digital-tool-for-landlords-measures-potential-tenants-kindness-cleanliness-1.3837486

  "It's very hard to see how information that is disclosed on Facebook, Instagram or Twitter would be related to a tenant suitability decision"

  Acting Deputy B.C. Privacy Commissioner Bradley Weldon.

From the cryptographic community

- Some people work for a small number of big players
  - Governments
  - Banks
  - Telcos
- Some people work for a big number of small players
  - Email encryption (PGP)
  - Onion routing (TOR)
  - Distributed ledgers (Bitcoin)

Missing piece in AI

- Plenty of freely available **frameworks**
- Not so many freely available **systems**
- Lack of training data
  - This is where Open Data can help
- Lack of an interested community
  - This where you can help

Outline

Project in one slide

- Open Data: governments releasing large quantities of unstructured information
  - Use IE to fulfill Open Data mission of increased awareness and transparency
- Full end-to-end systems solving IE problems:
  - On data that can be shared
  - On problems of public interest

Data and problems

- Quebec data
  - FIOA request by then Montreal Gazette journalist Roberto Rocha
    - 28,376 decisions
    - 112Mb
    - 458,190 lines
    - 2,456,420 words
    - 87 words per decision (average)
    - 25k pages if printed
  - NEQ: Enterprise Registry of Quebec (CC-BY-NC)
    - 2,088,934 companies
- Argentina/Costa Rica data: a decade of parliament proceedings

Solutions

- Octoroy
  - Proceedings of the Executive Committee in the cities of Montreal and Laval
  - French
  - Focus on finding which company got what money for what reason
- Vozyvoto
  - A study of group participation (both in terms of speaking and being addressed by other speakers) in the proceedings of government assemblies.
  - Currently female participation, to study the impact of gender quotas in Argentina

What it is, what is it not

- Focus on **high quality extraction** from **highly paradigmatic texts**
  - Very, very laborious
  - Plenty of industrial applications / interest
- Not directly related to
  - General news / Twitter extraction
    - Many of the techniques apply but the error rate is much higher
  - No Open IE (relations signaled by lexical items)
    - Interesting idea, but require extra (human) work for many traditional applications

Octroy

- Problem
  - Identify which decisions authorize (French *octroyer*) payment to companies
  - For these decisions, extract company name, amount and reason
  - Link company to the NEQ registry
- Types
  1. Enterprise
  2. Amount
  3. Reason
- Relation
  1. Contract

Future

- New problems
  - Local to BC / Vancouver?
- New contributors?

About Pablo

- Originally from Cordoba, Argentina
- Columbia University
  - Doctoral Dissertation: "Indirect Supervised Learning of Strategic Generation Logic", defended Jan. 2005.
- IBM Research Watson
  - Deep QA - Watson - Jeopardy! Show
- In Montreal from 2010-2016
  - Two semesters teaching at Cordoba University (NLG / ML on large datasets)
- Bootstrapping a NLG company (Textualization Software Ltd.) in Vancouver, Canada
- Writing a textbook on Feature Engineering

IE and me

- Started my PhD back in 1999 working on GeneWays
  - A multidisciplinary IE pipeline for genomics
- Did my PhD in Natural Language Generation
  - Intelligence domain (IE from news)
- At IBM: Enterprise Search competition (2006)
  - Expert search (expert detection and linking)
- After IBM worked on two IE projects
  - Real Estate contracts in French
  - Technical support from Web pages

Outline

A methology for IE

- Hybrid IE
  - Combining ML with custom programs (rules)
- Minimize annotation / development time
- Focus on evaluation
- Split the data into batches (of about 30 documents)
- At each step:
  - Train a model / tweak rules on existing data
  - Annotate a new batch of fresh data
    - Active learning can help here to select an "useful" subset of the fresh data
  - Evaluate the full system on the new annotations
  - If the results are good enough, finish, otherwise keep annotating

Specificity and Complexity Performance Tradeoff

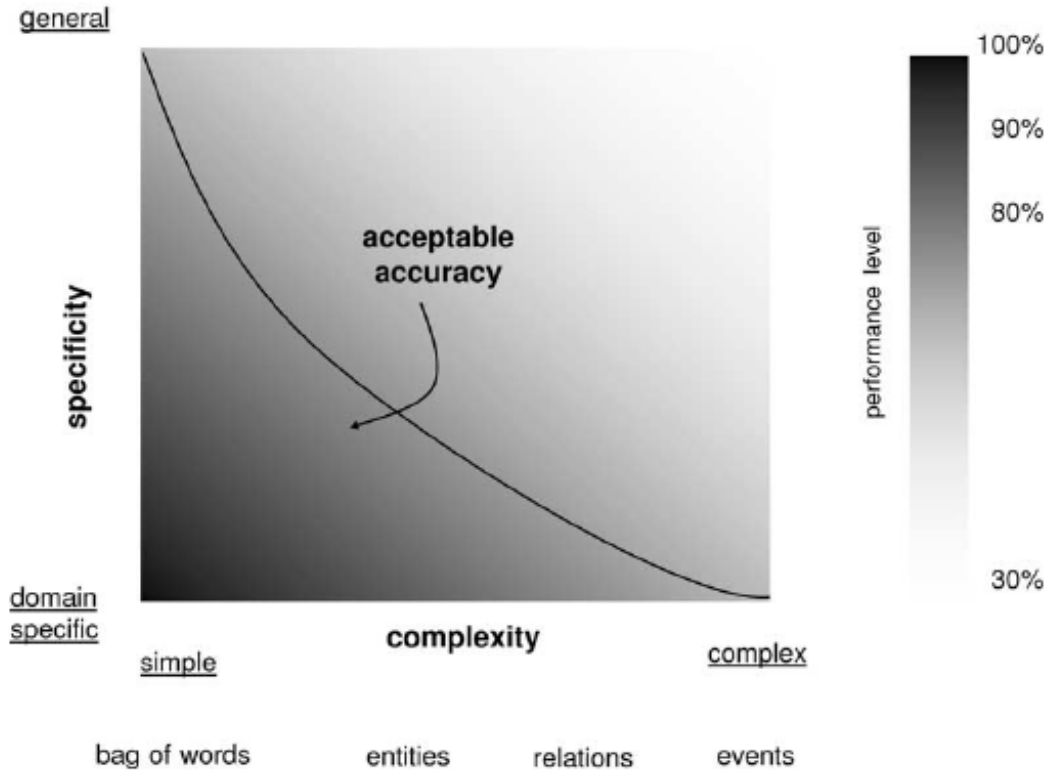

Figure 1 of Cunningham, H (2006)

Octroy Components

1. Collection reader (runs from the DB and brings up the date, too)
2. Relevant text segments (RuTA)
3. Sentence splitting (OpenNLP)
4. Tokenization (OpenNLP)
5. Amount annotation (RuTA)
6. Enterprise dictionary (using NEQ dictionary)
7. CRF (for Enterprise and Reason)
8. Postprocessing (RuTA)
9. Entity linking (to the NEQ ID)
10. Event extraction (populates the .cz ontology)
11. CAS Consumer (RDF writer)

Initial Data Analysis

- Data Release Schedule:
  - randomize document IDs, pick 30% as development set
  - 8,513 as devset
- Take the first 36 and analyze them by hand
  - 18 contains contracts (50%)
- Issues:
  - Approving money transfer to another gov't org (not a contract)
  - Contracts with no amount (kept)
  - Subsidies (kept as contract, the reason is subsidy)
  - When a company *pays* the gov't it has all the information as when the *gov't* pays a company.

Examples

Contrat de construction
CM Montréal-Nord , Bureau du directeur d'arrondissement - 1063602020
Octroyer un contrat de 1 328 000,65 $, taxes incluses à TGA Montréal inc. pour des travaux de réfection d'égout, d'aqueduc, de trottoirs et de pavages sur les avenues Drapeau, Éthier et Patricia (Contrat No 755)

Examples (cont.)

CE-2011/4460
CONTRAT - SOUMISSION «OS-ING/2011-041» ÉQUATION GROUPE CONSEIL INC.
RÉSOLU À L'UNANIMITÉ: que la soumission «OS-ING/2011-041» déposée par la firme Équation groupe conseil inc. concernant les services professionnels d'ingénieurs-conseils pour la préparation des plans et devis ainsi que pour les services durant la construction des travaux d'aménagement d'un terrain de tennis double au parc des Nénuphars, prévus au règlement L-11796, soit acceptée et qu'à cette fin, la firme susdite prépare les plans, devis et documents de soumissions au montant de 31 317,07 $; que les honoraires soient calculés conformément aux dispositions de la soumission «OS-ING/2011-041»; que le Greffier ou la Greffière adjointe retourne les garanties qui accompagnaient les soumissions non retenues. (C/T: 1207248) (Réf: 12-2)

Examples of Other Documents

CE-2011/4502
SOUMISSIONS «OS-27912» REJETÉES
RÉSOLU À L'UNANIMITÉ: que toutes les soumissions reçues portant le numéro «OS-27912» concernant l'acquisition et la mise en place d'un progiciel pour la gestion des heures et des horaires de travail (PPHT) soient et, par la présente, sont rejetées; que le Greffier ou la Greffière adjointe retourne les chèques ou garanties qui accompagnaient les soumissions. (Réf: 26-28)

Examples of Other Documents (cont.)

Immeuble - Aliénation
CE Mise en valeur du territoire et du patrimoine , Direction stratégies et transactions immobilières - 1074501006
Approuver un projet d'acte par lequel la Ville vend à Mark C. Moore, propriétaire des immeubles sis aux 3616-3652, rue Notre-Dame Ouest, un terrain situé dans l'arrondissement Le Sud-Ouest au sud-est de la rue Notre-Dame et au nord-est de la rue Bourget, constitué des lots 3 916 775 et 3 916 776 du cadastre du Québec, aux fins d'assemblage, d'une superficie totale d' environ 99,1 mètres carrés, pour le prix de 9 600 $, plus les taxes, si applicables, le tout sujet aux termes et conditions stipulés au projet d'acte.
Territoire(s) concerné(s) : Le Sud-Ouest District(s) : Saint-Henri - Petite-Bourgogne - Pointe-Saint-Charles

A Quick Perl Baseline

- Analysis of 36 documents
- 50 lines of perl
- Only documents with an amount and a company
- Only companies that end in Inc. (ignoring case)

A Quick Perl Baseline (cont.)

```
5   my@doc=<STDIN>;
6   chomp(@doc);
7   my$doc=join("␣",@doc); # one line, return characters transformed into spaces
8
9   # clean up header
10  if($doc=~ m/RÉSOLU À L'UNANIMITÉ\:/){
11  ␣␣␣␣$doc␣=~␣s/.*RÉSOLU␣À␣L'UNANIMITÉ://;
12  }
13
14  my@amounts = $doc =~
15    m/(\d?\d?(?:\s?|\.?)\d{3}(?:\s?|,?)(?:\d{3})?(?:\,\d{2})?\s?\$)/g;
16  if(!@amounts){ # no amount, bail-out
17      print "0\n"; exit;
18  }
```

A Quick Perl Baseline (cont.)

```
19  # has amount
20  if($#amounts){ # more than one amount, bail-out
21      print "0\n"; exit;
22  }
23  # got the amount, now find the company
24  my$amount = pop @amounts;
25
26  # is there a inc. ?
27  my@incs = split(/\sinc\./i, $doc);
28  if(!$#incs){ # no company, bail-out
29      print "0\n"; exit;
30  }
```

A Quick Perl Baseline (cont.)

```
32  # let's focus on the first one, we know $incs[0] ends in \sinc.
33  my$company = $incs[0];
34  # trim as much as possible
35  $company=~ s/.*firme//i;
36  $company=~s/.*à//i;
37  $company=~s/.*\spar\sle//i;
38  $company=~s/.*\sla\scompagnie\s//i;
```

A Quick Perl Baseline (cont.)

```
40  # reason is a crapshoot
41  my$reason="";
42  if($doc =~
43   m/((du\scontrat\sde)|(requis\spour)|(concernant)|
44  (\Q$amount\E\spour)|(\Q$company\E\s[Ii][Nn][Cc]\.\spour))/){
45      ($reason) = $doc =~ m/(?:(?:du\scontrat\sde)|(?:requis\spour)|
46  (?:concernant)|(?:\Q$amount\E\spour)|
47  (?:\Q$company\E\s[Ii][Nn][Cc]\.\spour))\s(.*)/;
48      # trim aggressively
49      $reason =~ s/(\,|\.|\;).*//;
50  }
51
52  print "1\t$amount\t$company␣inc.\t$reason\n";
```

Baseline Evaluation

- On the same 36 documents were it was programmed, at the document level by hand:

		System
Annotated	tp: 6	fn: 10
	fp: 0	tn: 20

  - precision = 6 / 6 = 1.0
  - recall = 6 / 16 = 0.375
- but partial reason in one and missing reason in another
  - 4 fn due to lack of amount, 6 due to lack of inc.

Baseline Evaluation (cont.)

- On 32 new documents, at the document level, by hand:

		System	
Annotated		tp: 5	fn: 9
		fp: 0	tn: 18

  - precision = 5 / 5 = 1.0
  - recall = 5 / 13 = 0.38
- but only 2 perfect, 1 wrong company and no reason, 1 no reason and 1 reason is whole doc
  - 2 fn due to lack of amount, 7 due to lack of inc.

IE Subtasks

1. Entity detection (/ linking)
2. Event detection
3. Frame building

Techniques:

- Regular Expressions
- Markov Sequence tagging
- Anchor-based rules
- Conditional Random Fields and beyond

What are Named Entities?

- At its core, proper names
- Nowadays generalized to nouns and multi-word expressions within a semantic class
  - 200 categories including "color" which contain common nouns
- Useful outside IE
  - Reduce the vocabulary space, instead of every name of every person have a token "NAME_OF_PERSON"

Some Example NEs

[Fred Flintstone]$^{\text{person}}$ was named [CTO]$^{\text{position}}$ of [Time Bank Inc.]$^{\text{organization}}$ in [2031]$^{\text{date}}$. The [next year] [he] got married and became [CEO]$^{\text{position}}$ of [Dinosaur Savings & Loan]$^{\text{organization}}$.

from Grishman (2012)

NE in the Context of IE

- NEs are the entries in the DB
- Events are the DB schema

Named Entity Recognition Techniques

- Three techniques
  - Regular Expressions
  - Lists of names (gazzetteers)
  - Machine learning
    - Word Sense Disambiguation
    - Semi-supervised

Regular Expressions

- Regular Expressions are a **succint way to encode an automaton** that accepts a regular language
- Constructs:
  - literal (e.g. /RÉSOLU/)
  - character class (e.g., /[0-9]/)
  - quantifiers (e.g., /,?/)
  - groups (e.g., /([0-9][0-9][0-9])/)

RE: Literal

- **Literals are characters or sequences of characters** that need to be matched verbatim
  - In perl code from the baseline: /(concernant)/
- Characters that have a meaning in the RE language (e.g., '?') need to be escaped (e.g., '\?')
  - In Java you will need to double escape (e.g., "\\?")
- Special quotation to escape unknown sequences: \E ... \Q
  - In perl code from the baseline: /\Q$amount\E\spour/

RE: Character Classes

- Succint way to describe a **set of characters**
  - Either by listing all members (e.g, /[xyz]/)
  - Or by using a range (e.g., /[0-9]/)
- There are also patterns for most common sets
  - /\d/ for **digits**
  - /\s/ for **white space**
  - special class /./ that matches **any character**

RE: Quantifiers

- Extend the smaller regular class recursively

?	one or nothing
*	nothing, one or more
+	one or more
{n,m}	at lest n, at most m

- From perl baseline /.*RÉSOLU À L'UNANIMITÉ:/

RE: Groups

- **Concatenate** other regular expressions
  - /concernant?\spour/ means the t is optional!
  - /(concernant)?\spour/ means concernant is optional
- The regular expressions inside a group could be of any complexity, including groups
  - /(concertnant\s)?pour/
- By default groups are capturing which means the match is returned by the system
  - Non-capturing groups are indicated with **?:**
  - /(?:concernant\s)?pour/

RE: Alternatives

- Indicates **two or more** regular expressions could be matched
  - /((du\scontrat\sde)|(requis\spour)|(concernant)/
  - Character classes are a succint way to representing many alternatives
- Watch out the need for grouping
  - /du\scontrat\sde|requis\spour/ means (du\scontrat\sd)[er](equis\spour) which is not what you want

Amount RE

- /(\d?\d?(?:\s?|\.?)\d{3}(?:\s?|,?)(?:\d{3})?(?:\,\d{2})?\s?\$)/
- (\d?\d? // two digits (optional)
  - (?:\s?|\.?) // a space or a period (optional)
  - \d{3} // three digits (required)
  - (?:\s?|,?) // a space or a period (optional)
  - (?:\d{3})? // three digits (optional)
  - (?:\,\d{2})? // an optional comma followed by two digits
- \s?\$) // an optional space with a required dollar sign

NE Detection

- Dictionary-based
- WSD
- Semi-supervised

Dictionary-based (Gazetteers)

- Gazzetteers and their problems
  - Spurious matches
  - In Octroy, "La Firme" was in some moment the name of a company
    - Same with "CE"
- Need annotated corpus for evaluation, otherwise more rules hurt performance

Simple Rules

- All capitalized sequences that end in "Inc." are companies
  - Work well as a starting point
- Need annotated corpus for evaluation, otherwise more rules hurt performance

Why Rules?

- Understandable
  - Debuggable
- Continuous Quality Improvement

Key: Rule Language

- Two approaches:
  1. Regular Expressions
  2. Anchor-based

Anchor-based Rules for IE

- Alternative to transducing the whole sequence of characters / tokens / annotations
- Identify key elements that are rare enough and reliable as a start for processing
  - Anchors
- Structure the processing around further conditions around these elements
  - The context around the anchor can then be modeled using REs

Some Rules

- CW+ @CompanyTrailing{->MARK(Company,1,2)};
- "la" "firme" W+{-CONTAINS(CompanyTrailing)} CompanyTrailing{->MARK(Company,3,4)};
- Company "concernant" ANY[2,50]{-REGEXP("[.,;]")->MARK(Reason)};

Why Statistical IE?

- Cost reduction
- Measurable quality
- Learning analog
- Generalize over training

Type of Models

- Maximum Entropy
- Conditional Random Fields
- Generalized Graphical Models
- Deep Learning

WSD-based

- The context around an occurrence determines its function
- If we can segment the text around likely NEs, we can then use the context to determine its type

Sequence Tagging (IOB)

- Given n-tags, create $2n+1$ classes:
  - B-tag: this word starts a tag
  - I-tag: this word is inside a tag
  - O: this word is outside all tags (background model)
- Learn a classifier that goes from features around a word to these classes

IOB Example

- From WNUT NER competition
  - tonite O
  - " O
  - running B-tvshow
  - wit I-tvshow
  - mjd I-tvshow
  - " O
  - live O
  - from O
  - 7- O
  - 9pm O
  - eastern O
  - sirius B-company
  - 211 O
  - . O

WNUT NER 2016 baseline python system

- Workshop on Noisy User-generated Text (W-NUT) NER:
  - `https://github.com/aritter/twitter_nlp/tree/master/data/annotated/wnut16`
  - Training data (annotated tweets) → train.feats:

O w[-1]=*planning* w[0]=*the* w[-1]|w[0]=planning|the w[0]|w[1]=the|next DICT=tv.tv_program DICT=people.person word=the word_lower=the
O w[-1]=*the* w[0]=*next* w[-1]|w[0]=the|next w[0]|w[1]=next|Disney DICT=tv.tv_program DICT=people.person word=next word_lower=next prefix=n prefix=ne prefix=nex suffix=t suffix=xt
B-facility w[-1]=*next* w[0]=*Disney* w[-1]|w[0]=next|Disney w[0]|w[1]=Disney|World DICT=tv.tv_program DICT=people.person word=Disney word_lower=disney prefix=d prefix=di prefix=dis suffix=y suffix=ey suffix=ney INITCAP INITCAP_AND_GOODCAP
I-facility w[-1]=*Disney* w[0]=*World* w[-1]|w[0]=Disney|World w[0]|w[1]=World|trip DICT=tv.tv_program DICT=people.person word=World word_lower=world prefix=w prefix=wo prefix=wor suffix=d

Sequence Tagging

- The main problem for applying traditional ML approaches to sequence tagging is the variable size of the input.
- P(first word being of class Company | first word is Disney and second word is Channel) << P(first word being of class Company | first word is Disney and second word is Pictures)
- Markov assumption:
  - The value at time t is only dependent of the value at times t-1, ..., t-k (where k is the **order** of the Markov model)
- Using the Markov assumption we can then train models to make local + context (of order k) decisions.

Using the whole context

- The models describe before make **local decisions**
- For more complex problems requiring improved results we want to make **global decisions**
  - Conditional Random Fields
    - Work well with small training data
  - Recurrent neural network nodels (sequence-to-sequence) using LSTMs
    - State of the art for large datasets

Outline

How to make IE4OpenData useful yet understandable?

- Information Extraction is not a trivial problem
- Solving it well enough to obtain useful results needs a lot of tweaking and a certain level complexity
- That level of complexity seems to be too high to attract new contributors

ElasticSearch vs. Whoosh

- If you have to index a set of documents and search over them, you can use Lucene (or its python equivalent, Whoosh)
  - It gives you the pieces to build a search engine to suit your needs
  - Sourcing documents and storing them as you please
  - Building queries using a variety of methods
- Alternatively, you can deploy ElasticSearch
  - Store the documents in a certain way
  - Index them in a certain way
  - Only query them in a certain way
- But ElasticSearch explains how to do useful things on its README, that is not possible with Whoosh

Readmification

- It might be the case that new technology more complicated that what it fits in a README file is never going to be popular?
- Writing a compelling README for the IE4OpenData project
  - What can IE4OpenData do for you?
  - What it is the shortest path to achieve this?

Conclusions

- The IE4OpenData project has been useful to me so far in terms of teaching
- Haven't realized its potential use at large
- Contributors welcome
- Trend: reduced textual open data