Machine Learning IRL

Olivia Buzek – IBM Watson

Twitter: @oliviadoesnlp

Who am I?

- Machine Learning Engineer at IBM Watson
  - Natural Language Understanding product
  - 9 different ML-based APIs in 5+ languages
    - e.g. Sentiment Analysis, Keyword Extraction, Named Entity Recognition
  - Team of 12!
- My role:
  - Porting existing APIs into yet more languages
  - Big wins in the last two years: Spanish and Korean

What do I know?

- Here to tell you…
  - How I got into the field, and to my job at Watson,
  - What routes are possible into machine learning roles,
  - What sort of projects I work on, and
  - How we make machine learning decisions in an industry setting

- Please feel free to ask questions, I’m here for your knowledge more than anything else!

How did I get here?

- University of Maryland undergrad
  - Computer Science and Linguistics
  - Took 2 machine learning classes while there
- Johns Hopkins University grad school
  - Computational Linguistics program
  - Two years of classes in text processing, machine learning, etc
- Forward Deployed Engineer at Palantir
  - Built Java frontends for data processing

Other paths…

- I’ve seen…
  - Physics PhD’s
  - Neuroscience undergrads
  - ”Data Science” course grads (e.g. from Galvanize, Coursera)
- People who end up in these jobs typically:
  - Try to keep up with the field (blogs, papers, following people on Twitter)
  - Like finding patterns in data
  - Have some sort of math / analytical background, and some sort of programming

Sidebar: Data Science vs Machine Learning

- Several names exist in the field:
  - Data Scientist
  - Machine Learning Scientist / Engineer
  - Deep Learning Specialist
  - Data Engineer
  - Business Analyst
- Jobs vary a lot in responsibilities!
- ML is still a fairly young field
- Some companies are doing all the latest techniques; others are fairly early - apply and see what happens
- Having done Machine Learning in school is still pretty rare

Sidebar: Data Science vs Machine Learning

	Machine Learning	Data Science
Skills Required	• Data cleaning and processing • ML engines: TensorFlow, Keras, Torch • Knowledge of common ML models (logistic regression, SVMs, neural networks, etc)	• Data cleaning and processing • Statistical toolkits: R, MATLAB • Data and statistical analysis
Customer Requests	• Can you build me a system that will constantly answer questions for me? • Build and iterate on a model	• Based on this data I've collected, what should I do next? • Analysis of a specific data problem with decisions

Note: This isn't foolproof! Companies don't always know what they're hiring for. Ask when you consider a job what the core responsibilities will be!

IBM Watson

- Originally known as “that supercomputer that won *Jeopardy!*”

- Today, Watson is a suite of analytical tools accessible via API
  - Personality Insights
  - Natural Language Understanding
  - Tone Analyzer
  - Conversation
  - Language Translator
  - Speech Recognition
  - …

What I do

- NLU is one of Watson's most-used products
- Global demand for automatic text processing for business
- Watson is heavily "cognitive" – meaning we use a lot of neural networks and deep learning to solve problems (but not exclusively)
- We have to expand to new languages
- My biggest projects: Spanish and Korean language expansion

Watson's Natural Language Understanding (NLU) API

Customer problem:
"I have 1,000 blog posts that I've collected about concerts. I want to find up-and-coming bands. What can Watson extract?"

Demo: Pitchfork article– Björk, "Blissing Me"
URL: https://pitchfork.com/reviews/tracks/bjork-blissing-me

Watson NLU Demo on Bluemix –
https://natural-language-understanding-demo.mybluemix.net

Learn more about NLU: https://www.ibm.com/watson/services/natural-language-understanding/

The ML Lifecycle

- Idea: What sort of problems are worth solving? How do we figure out how to solve them?
- Data: How do we collect it? What considerations do we have to make when we collect it?
- Metrics and evaluation: How do we decide what metrics are appropriate for the task? Where do we get test data? Does test data always look like training data? Where do we get annotations?
- Programming: What do we program in? Why do we decide certain algorithms are better than others?
- Production systems: What other considerations do we have to make? (load testing? unit testing?)

The ML Lifecycle: Idea

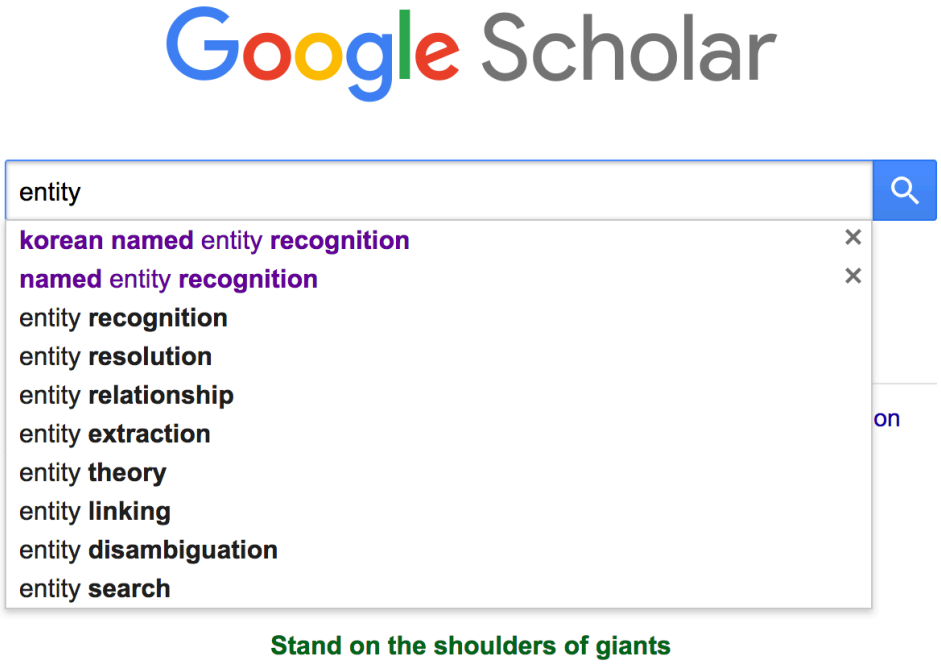
- Idea:
  - What sort of problems are worth solving?
  - How do we figure out how to solve them?

The ML Lifecycle: Idea

- Most things have been done before – literature search should be one of your first steps
- Figure out what your task is called!
- Our APIs:
  - Concepts = Concepts Extraction
  - Categories = Categorization
  - Keywords = Keyword Extraction
  - Entities = Named Entity Recognition
  - Sentiment = Sentiment Analysis

The ML Lifecycle: Idea

- Even a simple Google Scholar (or ArXiV) search should help you understand what the possibilities are



The ML Lifecycle: Idea

- You'll find lots of results
- Almost every paper will have terms you don't understand!
- That's normal, especially when you start 🙃
- Survey papers are a good place to get going – they'll reference everything else being done in the literature
- Move on to find one paper

A survey of **named entity recognition** and classification [PDF] nyu.edu

D Nadeau, S Sekine - Lingvisticae Investigationes, 2007 - jbe-platform.com

This survey covers fifteen years of research in the **Named Entity Recognition** and Classification (NERC) field, from 1991 to 2006. We report observations about languages, **named entity** types, domains and textual genres studied in the literature. From the start,

☆ 99 Cited by 1439 Related articles All 21 versions

The ML Lifecycle: Idea

- No pre-existing system?
  - Try to replicate a paper that already exists, and gets high performance on a problem similar to yours

Computer Science > Computation and Language

Boosting Named Entity Recognition with Neural Character Embeddings

Cicero Nogueira dos Santos, Victor Guimarães

(Submitted on 19 May 2015 (v1), last revised 25 May 2015 (this version, v2))

Most state-of-the-art named entity recognition (NER) systems rely on handcrafted features and on the output of other NLP tasks such as part-of-speech (POS) tagging and text chunking. In this work we propose a language-independent NER system that uses automatically learned features only. Our approach is based on the CharWNN deep neural network, which uses word-level and character-level representations (embeddings) to perform sequential classification. We perform an extensive number of experiments using two annotated corpora in two different languages: HAREM I corpus, which contains texts in Portuguese; and the SPA CoNLL-2002 corpus, which contains texts in Spanish. Our experimental results shade light on the contribution of neural character embeddings for NER. Moreover, we demonstrate that the same neural network which has been successfully applied to POS tagging can also achieve state-of-the-art results for language-independet NER, using the same hyperparameters, and without any handcrafted features. For the HAREM I corpus, CharWNN outperforms the state-of-the-art system by 7.9 points in the F1-score for the total scenario (ten NE classes), and by 7.2 points in the F1 for the selective scenario (five NE classes).

The ML Lifecycle: Idea

- People also share a lot of information on blogs these days

- Ex: https://guillaumegenthial.github.io/sequence-tagging-with-tensorflow.html

- Still structured like a paper, and often easier to replicate!
- Code for these is frequently open-sourced, because data is considered the “secret sauce”

The ML Lifecycle: Idea

- You can also just try things!
- If you want to play around from scratch, you know an algorithm, you have a metric in mind, and you have some data – have at it!
- Literature can help you figure out how well your approach does before you try it – sometimes.
- Half-science, half-art, and over time people will often gravitate towards methods they know, until something new comes along

The ML Lifecycle: Data

- Data:
  - How do we collect it?
  - What considerations do we have to make when we collect it?
  - Where do we get annotations?

The ML Lifecycle: Data

- What I know best is text data
- Where do we get the data?
  - News crawls – news.google.com is a great source for articles
  - Wikipedia – you can download the whole thing!
    - https://github.com/attardi/wikiextractor
  - Dbpedia – the metadata analogue to Wikipedia
    - Also has open sourced downloads – essentially a database of links from place to place
  - LDC corpora
    - https://catalog.ldc.upenn.edu/ - UPenn aggregates as many text corpora as they can here, in standardized formats
    - Most universities, and some companies, have memberships – most of the corpora can also be purchased for USD $35.00
    - Includes some standard corpora that already come pre-annotated, for common classes of problems

The ML Lifecycle: Data

- Learn to check licenses on data, and licenses on open source software
- You may not know which licenses make things usable for you or not at first!
  - TLDR Legal can help you out: https://tldrlegal.com/

The ML Lifecycle: Data Annotation

- How do you get annotations?
  - DIY annotations – have an annotation party with your friends!
    - Free, but slow
  - Crowdsourcing – Amazon's Mechanical Turk, CrowdFlower, DefinedCrowd, among others
    - Less expensive than trained annotators; pretty fast
  - Trained linguists / annotators
    - Slow, expensive; sometimes it's what you need though

The ML Lifecycle: Data Annotation

- How complex can annotation be?

<u>Your Entity Types:</u>
Location
GeographicFeature
Organization
Company
Facility

"I flew to **Hawaii** over winter break."

"I flew to the **Hawaiian Islands** over winter break."

"I hung out at the **Platte River Brewing Company** last night."

"I attend the **University of Colorado** in **Boulder**."

"I studied at the **Boulder Public Library** last night."

Our annotation guidelines for our types are **35** pages long!!
The more clear and specific your guidelines, the better your data.

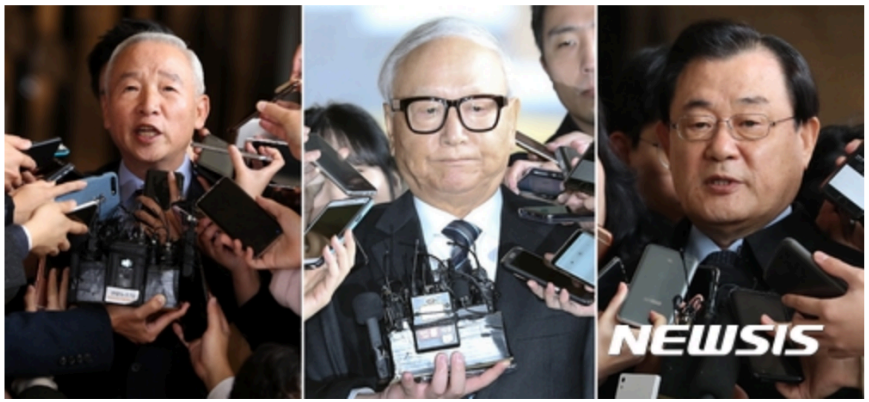
The ML Lifecycle

- Now imagine Korean…

朴국정원장들 엇갈린 운명…남재준·이병기 구속,

[뉴시스] 입력 2017.11.17 01:45

0



재직 시 특활비 청와대 정기적 상납 혐의

法 "이병호, 증기인멸 염려 없다"영장 기각

【서울=뉴시스】오제일 기자 = 박근혜 정부 시절 국가정보원 특수활동비를 청와대에 정기적으로 상납한 혐의를 받고 있는 남재준·이병기 전 국정원장이 17일 구속됐다. 이병호 전 국정원장은 영장이 기각되면서 구속 위기를 면했다.

전날 박근혜 정부에서 국정원장을 지낸 3인에 대한 영장심사를 진행한 권순호 영장전담

The ML Lifecycle: Data Annotation

- Annotations can also be extracted from structured data
- Polyglot method – learning from structured resources like Dbpedia
- Uses the links in Wikipedia to guess at entity types

Al-Rfou et al., 2015: POLYGLOT –NER: Massive Multilingual Named Entity Recognition

The ML Lifecycle: Data Annotation

Barack Obama

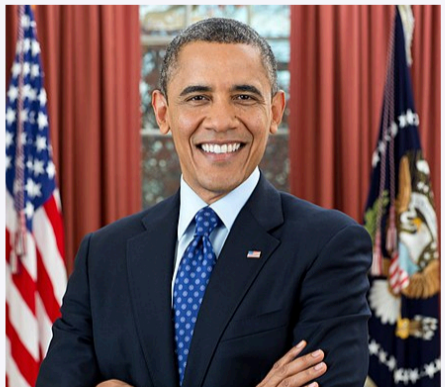
From Wikipedia, the free encyclopedia

"Barack" and "Obama" redirect here. For other uses, see Barack (disambiguation) and Obama (disambiguation).

Barack Hussein Obama II (US: /bəˈrɑːk huːˈseɪn oʊˈbɑːmə/ (listen) *bə-**RAHK** hoo-**SAYN** oh-**BAH**-mə*;[1][2] born August 4, 1961) served as the 44th President of the United States from 2009 to 2017. The first African American to assume the presidency in U.S. history, he previously served in the U.S. Senate representing Illinois from 2005 to 2008 and in the Illinois State Senate from 1997 to 2004.

Obama was born in 1961 in Honolulu, Hawaii, two years after the territory was admitted to the Union as the 50th state. Raised largely in Hawaii, Obama also spent one year of his childhood in Washington State and four years in Indonesia. After graduating from Columbia University in 1983, he worked as a community organizer in Chicago. In 1988 Obama enrolled in Harvard Law School, where he was the first black president of the *Harvard Law Review*. After graduation, he became a civil rights attorney and professor, and taught constitutional law at the University of Chicago Law

Barack Obama



Al-Rfou et al., 2015: POLYGLOT –NER: Massive Multilingual Named Entity Recognition

The ML Lifecycle: Metrics and Evaluation

- Metrics and evaluation:
  - How do we decide what metrics are appropriate for the task?
  - Where do we get test data?
  - Does test data always look like training data?

The ML Lifecycle: Metrics and Evaluation

- Most tasks have standard metrics that go with it in the literature
  - Named Entity Recognition uses a modified F1 score
    - (though there's debate in the community as to whether that's the "right" way to score the task)
  - Sentiment Analysis can use F1, or Accuracy, depending on how you frame it
- Define your metrics before you start:
  - You want to know what "success" looks like before you begin
  - Otherwise, it's easy to get stuck on a quest for perfect scores, or at least driving them higher – that may not be warranted in your case!
- Define your test data before you start!
  - Often, people split training data 80% / 10% / 10% - train on 80%, use 10% while you're developing, and use the last 10% as benchmarks for release

The ML Lifecycle: Metrics and Evaluation

- When it comes time to put an ML product to market, we frequently do a final quality test on “wild” data – data we found that day
- This helps eliminate the “past” bias that’s inevitable in machine learning
- Say you train only on data from before 2008. What does your model know about current events?
  - Does your algorithm know about Donald Trump winning the 2016 election?
- For machine learning in real life, you have to have a plan for how you’re going to keep your system up to date

The ML Lifecycle: Programming

- Programming:
  - What do we program in?
  - Why do we decide certain algorithms are better than others?

The ML Lifecycle: Programming

- Language / framework comparisons and benchmarks
  - https://www.tensorflow.org/performance/benchmarks
  - https://github.com/jcjohnson/cnn-benchmarks
- You can also use Weka, or write from scratch!
- These days, it's most common to use open-source programming-based frameworks, unless there's a wheel you need to reinvent
- Well-known and used frameworks:
  - TensorFlow (Google)
  - Torch (Facebook)
  - Theano (Al-Rfou et al.)
  - Keras (lives on top of the above)

The ML Lifecycle: Programming

- Choosing an algorithm / model:
  - Decision trees are fun and easy to understand, but not usually the right model
  - Research comes in handy here – what are the most commonly used methods for your task?
  - Representation: what does one model represent that another doesn't?
  - Complexity and data size: Do you have enough data to warrant a complex model?
    - Logistic regression solves a whole lot of problems
    - So do SVMs
    - Neural networks are cool and fancy, but have their tradeoffs in speed and data needs
  - Type of problem: Classification, regression, sequence tagging (form of classification); supervised, unsupervised

The ML Lifecycle: Production Systems

- Production systems: What other considerations do we have to make?

The ML Lifecycle: Production Systems

- When you're dealing with real-life customers and trying to serve a machine learning model at scale, you have even more concerns
- Software engineering knowledge is really useful here!
- Consider things like:
  - Load tests – how many requests can your model handle, and how quickly?
  - Cost of compute – does your model run on CPU? Or does it require several GPUs?
  - How can you optimize the running speed?
  - How can you make sure your system only loads the model once?

That’s all, folks!

Back to your regularly scheduled Machine Learning class.

Feel free to come talk to me after