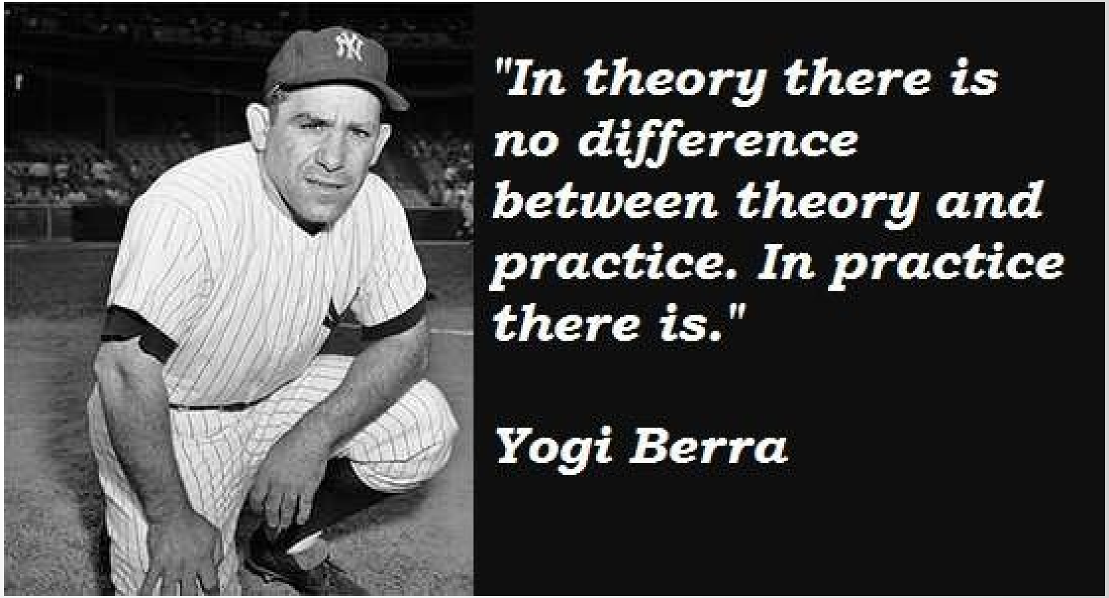
Winning together: Bridging the gap between academia and industry

Radim Řehůřek, Ph.D.
rare-technologies.com
@radimrehurek

PyData
Warsaw 2017



MSc: SVMs on bio data, 2005


MASARYK UNIVERSITY

Search

web pages | employees

About the university | Research and development | Applicants | Students | Alumni | Information sources | News | Partners

Masaryk University

About the university

- MU profile
- People at MU
- MU activities
- History
- University history
  - The MU Establishing Act (Act No. 50/1919)
  - Brief history
  - Timeline
  - Calendar of events
  - Deed of gift of Kounic Palace *(Cz)*
  - T.G.Masaryk
  - Auditorium Maximum (MU Great Hall)
  - Annual reports
- People in the history of the university
- University Awards

Brief history

zoom

Masaryk University was established in 1919, shortly after the creation of an independent Czechoslovak state. However, the decision to establish a university did not come in a sudden burst of revolutionary fervour. Rather, it represented the culmination of many years of effort on the part of Czech society, then in the process of developing rapidly on all fronts, to establish a second centre of national education and culture. The campaign was spearheaded by **Tomáš Garrigue Masaryk**, who – as early as in the 1880s – stressed the need for the greatest possible diversity in scientific and scholarly life, pointing out that the single then existing Czech university, i.e. Charles University in Prague, needed a counterpart within the country if it was do develop properly. For many years the task of establishing a second Czech university was one of Masaryk's main political priorities. He was not alone in his endeavours: the university question was taken up not only by professors and students at Charles University, but by the public at large. For decades, motions calling for the establishment of a second university were presented in the Reichsrat in Vienna and at the Moravian Diet in Brno.

In Moravia the movement for a second Czech university was linked to a campaign for the reopening of a Moravian

Search engines, NLP: 2007

PhD in 2011: NLP, scaling up topic modelling

Several open source libs

RaRe-Technologies / **gensim**

Unwatch 342 | Unstar 5,205 | Fork 2,091

Code | Issues 135 | Pull requests 20 | Projects 2 | Wiki | Insights | Settings

185 lines (141 sloc) | 13.1 KB

Raw | Blame | History

gensim – Topic Modelling in Python

build passing | release v2.3.0 | wheel yes | Mailing List | gitter join chat → | Follow 3k

Gensim is a Python library for *topic modelling*, *document indexing* and *similarity retrieval* with large corpora. Target audience is the *natural language processing* (NLP) and *information retrieval* (IR) community.

Features

- All algorithms are **memory-independent** w.r.t. the corpus size (can process input larger than RAM, streamed, out-of-core),
- **Intuitive interfaces**
  - easy to plug in your own input corpus/datastream (trivial streaming API)
  - easy to extend with other Vector Space algorithms (trivial transformation API)
- Efficient multicore implementations of popular algorithms, such as online **Latent Semantic Analysis (LSA/LSI/SVD)**, **Latent Dirichlet Allocation (LDA)**, **Random Projections (RP)**, **Hierarchical Dirichlet Process (HDP)** or **word2vec deep learning**.
- **Distributed computing**: can run *Latent Semantic Analysis* and *Latent Dirichlet Allocation* on a cluster of

RARE Technologies Ltd.

AUTODESK
DATASIFT
elevate
SIGNAL MARITIME
HEARST
amazon.com
SPORTS AUTHORITY.
Brainspace
Eva
Expert Virtual Agent
DynAdmic
SUWECO
rally
harvest.ai
People Ticker

2016: RARE Incubator, academic partnerships

RARE BRINGS INNOVATION FROM THE CLASSROOM TO THE BOARDROOM

We're the first to know about analytical advancements,
because we work with the researchers developing them.

RaRe's official, exclusive and ongoing partnerships with leading universities put us at the forefront of new developments in machine learning and give students access to experienced mentors, expert assistance and employment opportunities.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

DTU

plin

Students

We offer practical thesis topics in data mining and offer paid internships for ambitious and talented students worldwide.

More on Our Incubator

Universities

We mentor and review students' theses and work collaboratively in official industry partnerships for large consortial projects.

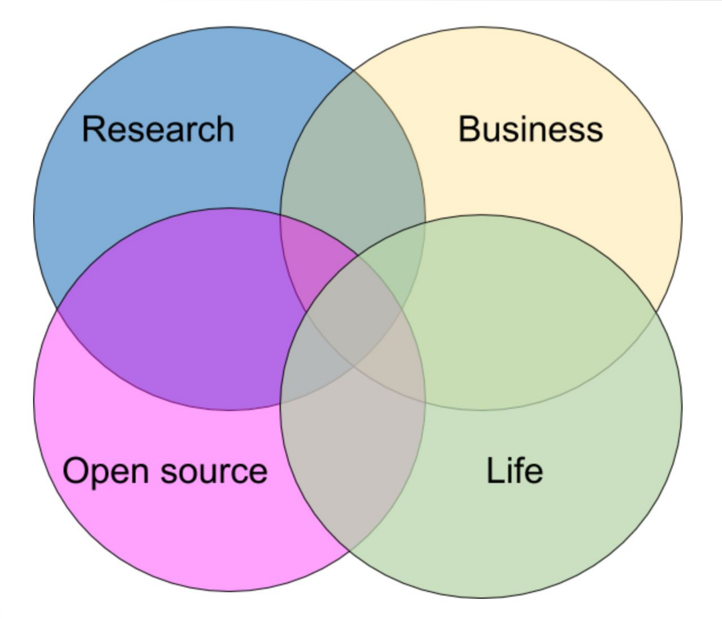
Have a Project?

Good data scientists are hard to find – so we help grow them.

East Asia since 2009

Academia vs industry friction points

1. Managing risk
2. Ownership & Sustainability



Friction point #1: Managing risk

Risk is **the** fundamental axis for a business

- Fear of new things destabilizing hard-won processes
- vs. fear of becoming obsolete.

Source of friction:

- business: wants everything **repeatable, replaceable, orderly**
- research (art, craft, ...): **unique, novel, creative**

Managing risk: Business horror, researcher's dream?

- Scariest thing to business: **magic opaque black-box** at the heart of your business.
- Aka "Computer Says No".
- Opposite of decreasing risk, repeatability.

Learn2Learn: Learn the Optimization Update Rule

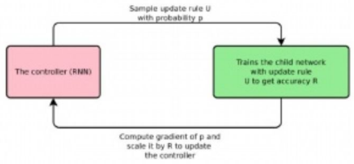


Figure 1. Overview of Neural Optimizer Search.

Optimizer	Final Val	Final Test	Best Val	Best Test
SGD	92.0	91.8	92.9	91.9
Momentum	92.7	92.1	93.1	$\boxed{92.3}$
ADAM	90.4	90.1	91.8	90.7
RMSProp	90.7	90.3	91.4	90.3
$[e^{\text{sign}(g)*\text{sign}(m)} + \text{clip}(g, 10^{-4})] * g$	92.5	92.4	93.8	93.1
$\text{clip}(\hat{m}, 10^{-4}) * e^{\hat{v}}$	93.5	92.5	93.8	92.7
$\hat{m} * e^{\hat{v}}$	93.1	92.4	93.8	92.6
$g * e^{\text{sign}(g)*\text{sign}(m)}$	93.1	92.8	93.8	92.8
$\text{drop}(g, 0.3) * e^{\text{sign}(g)*\text{sign}(m)}$	92.7	92.2	93.6	92.7
$\hat{m} * e^{g^2}$	93.1	92.5	93.6	92.4
$\text{drop}(\hat{m}, 0.1)/(e^{g^2} + \epsilon)$	92.6	92.4	93.5	93.0
$\text{drop}(g, 0.1) * e^{\text{sign}(g)*\text{sign}(m)}$	92.8	92.4	93.5	92.2
$\text{clip}(\text{RMSProp}, 10^{-5}) + \text{drop}(\hat{m}, 0.3)$	90.8	90.8	91.4	90.9
$\text{ADAM} * e^{\text{sign}(g)*\text{sign}(m)}$	92.6	92.0	93.4	92.0
$\text{ADAM} * e^{\hat{m}}$	92.9	92.8	93.3	92.7
$g + \text{drop}(\hat{m}, 0.3)$	93.4	92.9	93.7	92.9
$\text{drop}(\hat{m}, 0.1) * e^{g^3}$	92.8	92.7	93.7	92.8
$g - \text{clip}(g^2, 10^{-4})$	93.4	92.8	93.7	92.8
$e^g - e^{\hat{m}}$	93.2	92.5	93.5	93.1
$\text{drop}(\hat{m}, 0.3) * e^w$	93.2	93.0	93.5	$\boxed{93.2}$

Table 1. Performance of Neural Optimizer Search and standard optimizers on the Wide-ResNet architecture (Zagoruyko & Komodakis, 2016) on CIFAR-10. Final Val and Final Test refer to the final validation and test accuracy after for training for 300 epochs. Best Val corresponds to the best validation accuracy over the 300 epochs and Best Test is the test accuracy at the epoch where the validation accuracy was the highest.

Neural Optimizer Search using Reinforcement Learning,
Irwan Bello, Barret Zoph, Vijay Vasudevan, and Quoc Le. To appear in ICML 2017

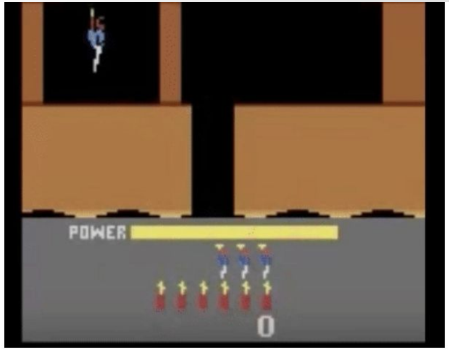
Managing risk: Take "SOTA" easy

mat kelcey @mat_kelcey · May 20

pretty much every paper i've ever read....

method	score
previous approach	good
our approach	almost as good
our approach + last minute hack	slighty better than good!

91 82 View pho



Be wary of non-breaking bugs: when we looked through a sample of ten popular reinforcement learning algorithm reimplementations we noticed that six had subtle bugs found by a community member and confirmed by the author. These ranged from mild bugs that ignored gradients on some examples or implemented causal convolutions incorrectly to serious ones that reported scores higher than the true result.

...except disagree with "hack" as pejorative!

Managing risk: The Mummy effect

jayant jain Oct 16 (1 day ago)

to [redacted], opensource

Hi [redacted]

I'm an engineer at RaRe Technologies and we are working on a Python implementation [redacted] described in your paper. We will publish it as open source into the Gensim repository.

First of all, thanks a lot for the well-explained paper, it was a great read, and the model described looks very useful.

I've set up some of the evaluation experiments - [redacted]. I've trained [redacted]. I've looked into the implementation in a fair amount of detail and it seems to match the model described in the paper exactly, with slightly different default values for some of the hyperparameters.

The embeddings do quite well on both these tasks. However we've been unable to reproduce the same numbers from the paper. The best spearman's score we've been able to achieve is 0.47, slightly lower than the 0.51 mentioned in the paper, and the best mean rank for the WordNet task around 50, significantly higher than the 3.83 in the paper.

To do a proper evaluation, using the same hyperparameters and evaluation settings as used in the paper would be ideal. So if you could help us out with some of our questions about the training/evaluation settings not mentioned in the paper, that would be great -

Training -

1. Initial and final learning rate for training and "burn-in", and if you're using a linearly decreasing learning rate
2. Number of epochs or stopping criterion, and whether this differs for the different evaluation tasks
3. Number of threads used
4. Train/test split ratio for link prediction on the WordNet data
5. Train/validation/test split for link prediction on the scientific collaboration datasets

Some clarifications on some ambiguities in the evaluation task would also be very helpful -

1. While using embeddings trained on [redacted], some words from HyperLex can have multiple senses in WordNet (and therefore multiple vectors). How is the choice of which sense/vector to use made?
2. Some words from HyperLex seem to be missing from the WordNet data (182/2163). Are these ignored in the evaluation?

Managing risk: Aggregate numbers

"The purpose of computation is insight, not numbers."

\- *Richard Hamming*

```
from gensim.models import Word2Vec

# define training data
sentences = ["hello world", "how is it going"]

# train model
model = Word2Vec(sentences=sentences, size=200, workers=4)

# ...use trained model in upstream task...
```

```
German JJ B-NP O
call NN I-NP O
to TO B-VP O
boycott VB I-VP O
British JJ B-NP O
lamb NN I-NP O
. . O O

Peter NNP B-NP O
Blackburn NNP I-NP O

BRUSSELS NNP B-NP O
1996-08-22 CD I-NP O

The DT B-NP O
European NNP I-NP O

1010 X X B-ADDRESS
N X X I-ADDRESS
MAIN X X I-ADDRESS
ST X X I-ADDRESS
28144 X X I-ADDRESS

416 X X B-ADDRESS
ST X X I-ADDRESS
MARKS X X I-ADDRESS
CT X X I-ADDRESS
PEORIA X X I-ADDRESS
```

Managing risk bridge #1:
Basic sanity checks

- unit tests (harmful!) utopia BUT:
- **concrete** logging and asserts instead of comments
  - sprinkle a few {random | head} data samples at various places along the data pipeline
- eyeball logs for anomalies
  - human brain still the best anomaly detector
  - does the data at each pipeline point match your expectations?

Managing risk RARE bridge #1: Basic sanity checks

```
class Dictionary(object):
    ...

    def __str__(self):
        sample_keys = list(itertools.islice(iterkeys(self.token2id), 3))
        return "Dictionary(%i unique tokens: %s%s)" % (len(self), sample_keys, '...' if len(self) > 3 else '')
=>
2017-08-25 09:45:39,441 : INFO : built Dictionary(12327 unique tokens: ['empirical', 'model', 'estimating']...)

Word2Vec(vocab=102, vector_size=300, alpha=0.025)
```

Cheap wins:

- catch word2vec vocab
- catch binary data in tokens

Managing risk bridge #2: Interactive demos

- Publications needed for citations, but times are changing.
- Blog posts, reproducible notebooks, visualizations, interactive web prototypes!
- **Guaranteed** to learn unexpected things about your system.
- “More eyes make all problems shallow”

Managing risk RARE bridge #2: Interactive demos

Bonus app

As before with finding similar articles in the English Wikipedia with Latent Semantic Analysis, here's a bonus web app for those who managed to read this far. It uses the word2vec model trained by Google on the Google News dataset, on **about 100 billion words**:

If you don't get "queen" back, something went wrong and baby SkyNet cries.
Try more examples too: "he" is to "his" as "she" is to ?, "Berlin" is to "Germany" as "Paris" is to ? (click to fill in).

man is to king as woman is to ?

Try: U.S.A.; Monty_Python; PHP; Madiba (click to fill in).

iPhone

Get most similar

Also try: "monkey ape baboon human chimp gorilla"; "blue red green crimson transparent" (click to fill in).

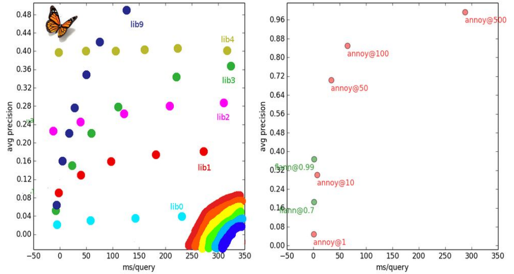
dinner cereal breakfast lunch

Which phrase doesn't fit?

So, which nearest-neighbour implementation is the best?

FLANN is spectacularly fast, but it's hard to say how it would fare on better accuracies.

On that note, let me say one more thing: getting to these results was about a **hundred times more painful than I had anticipated**. Lots of suffering. Things freezing, or not compiling, then compiling but failing tests, running out of memory, quirky idiosyncratic interfaces… And that's not counting the libraries that I had pruned outright last time. I really thought the landscape of open source high-dim k-NN implementations would be a lot merrier than this. I find it surprising, given how fundamental and well-researched the domain is academically.



Landscape of open source high-dim similarity libraries for Python. **Left:** What I imagined. **Right:** Reality, one-and-a-half survivors. Brutal shootout.

Friction point #2: Ownership & Sustainability

reddit MACHINELEARNING comments

1510

Discussion [D] Why can't you guys comment your fucking code? (self.MachineLearning)

submitted 3 months ago by didntfinishhighschoo x2

Seriously.

I spent the last few years doing web app development. Dug into DL a couple months ago. Supposedly, compared to the post-post-post-docs doing AI stuff, JavaScript developers should be inbred peasants. But every project these peasants release, even a fucking library that colorizes CLI output, has a catchy name, extensive docs, shitloads of comments, fuckton of tests, semantic versioning, changelog, and, oh my god, better variable names than `ctx_h` or `lang_hs` or `fuck_you_for_trying_to_understand`.

The concepts and ideas behind DL, GANs, LSTMs, CNNs, whatever – it's clear, it's simple, it's intuitive. The slog is to go through the jargon (that keeps changing beneath your feet - what's the point of using fancy words if you can't keep them consistent?), the unnecessary equations, trying to squeeze meaning from bullshit language used in papers, figuring out the super important steps, preprocessing, hyperparameters optimization that the authors, oops, failed to mention.

Sorry for singling out, but look at this - what the fuck? If a developer anywhere else at Facebook would get this code for a review they would throw up.

- Do you intentionally try to obfuscate your papers? Is pseudo-code a fucking premium? Can you at least try to give some intuition before showering the reader with equations?
- How the fuck do you dare to release a paper without source code?
- Why the fuck do you never ever add comments to you code?
- When naming things, are you charged by the character? Do you get a bonus for acronyms?
- Do you realize that OpenAI having needed to release a "baseline" TRPO implementation is a fucking disgrace to your profession?
- Jesus christ, who decided to name a tensor concatenation function `cat`?

502 comments share save hide give gold report

Implementing Latent Dirichlet Allocation - notation confusion

1

1

I am trying to implement LDA using the collapsed Gibbs sampler from http://www.uoguelph.ca/~wdarling/research/papers/TM.pdf

the main algorithm is shown below

Input: words $\mathbf{w} \in$ documents $\mathbf{d}$
Output: topic assignments $\mathbf{z}$ and counts $n_{d,k}$, $n_{k,w}$, and n_k
begin
randomly initialize $\mathbf{z}$ and increment counters
foreach *iteration* **do**
for $i = 0 \to N-1$ **do**
$word \leftarrow w[i]$
$topic \leftarrow z[i]$
$n_{d,topic}$-=1; $n_{word,topic}$-=1; n_{topic}-=1
for $k = 0 \to K-1$ **do**
$p(z=k|\cdot) = (n_{d,k} + \alpha_k)\frac{n_{k,w}+\beta_w}{n_k+\beta \times W}$
end
$topic \leftarrow$ sample from $p(z|\cdot)$
$z[i] \leftarrow topic$
$n_{d,topic}$+=1; $n_{word,topic}$+=1; n_{topic}+=1
end
end
return $\mathbf{z}$, $n_{d,k}$, $n_{k,w}$, n_k
end

Algorithm 1: LDA Gibbs Sampling

I'm a bit confused about the notation in the inner-most loop. n_dk refers to the count of the number of words assigned to topic k in document d, however I'm not sure which document d this is referring to. Is it the document that *word* (from the next outer loop) is in? Furthermore, the paper does not show how to get the hyperparameters alpha and beta. Should these be guessed and then tuned? Furthermore, I don't understand what the *W* refers to in the inner-most loop (or the beta without the subscript).

Could anyone enlighten me?

gibbs dirichlet-distribution topic-models

share cite improve this question

edited Sep 6 '13 at 17:45 ttnphns

asked Sep 6 '13 at 15:56 user1893354

Ownership & sustainability: The arXiv deluge

Used to be:

- Public scrutiny from low-volume peer reviews
- Publications high added value

Now:

- “Publish or Perish” crapshoot, flag-planting
- Twenty-seven percent of papers in the natural sciences are never cited.
  - fact http://onlinelibrary.wiley.com/doi/10.1002/asi.21011/abstract
- Only 1.6 people, on average, read a PhD thesis, and that’s including the author
  - joke (?)

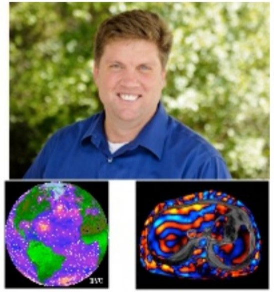
Ownership & Sustainability: Academic incentives for code & tools :(

> "Every great open source math library is built on the ashes of someone's academic career."

For example...

Travis Oliphant - CEO

- PhD 2001 from Mayo Clinic in Biomedical Engineering
- MS/BS degrees in Elec. Comp. Engineering
- Creator of **SciPy** (1999-2009)
- Professor at BYU (2001-2007)
- Author of **NumPy** (2005-2012)
- Started **Numba** (2012)
- Founding Chair of **Numfocus** / **PyData**
- Previous PSF Director



NumPy SciPy

Num FOCUS Open Code, Better Science

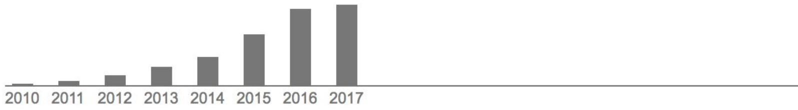
python

3

William Stein (SMC) SageMath June 11, 2016 32 / 33

Software framework for topic modelling with large corpora

Authors	Radim Rehurek, Petr Sojka
Publication date	2010
Conference	THE LREC 2010 WORKSHOP ON NEW CHALLENGES FOR NLP FRAMEWORKS
Pages	45--50
Publisher	University of Malta
Description	Abstract Large corpora are ubiquitous in today's world and memory quickly becomes the limiting factor in practical applications of the Vector Space Model (VSM). In this paper, we identify a gap in existing implementations of many of the popular algorithms, which is their scalability and ease of use. We describe a Natural Language Processing software framework which is based on the idea of document streaming, ie processing corpora document after document, in a memory independent fashion. Within this framework, we ...
Total citations	Cited by 687



http://wstein.org/talks/2016-06-sage-bp/bp.pdf

Ownership & sustainability bridge #1:
Less fire & forget

- “What am I looking at? Why is this important?”
  - Spend more effort on articulation of context, motivation, use-cases.
  - Blog: Do a layman version, without the acronyms and “obvious” assumptions.
  - Notebooks and interactive plots; legacy publication business ossified
  - Release a reference implementation (obviously)
- “Explain” = GOLD
  - Model interpretability
  - Getting the problem right >> SOTA
  - Real impact in understanding the goal, requirements, constraints, success metrics, data...

Ownership & sustainability bridge #1:
Less fire & forget

WordRank embedding: “crowned” is most similar to “king”, not word2vec’s “Canute”

PARUL SETHI / 2017-01-23 / GENSIM, STUDENT INCUBATOR

```
In [16]: model.most_similar('king')
Out[16]:
[(u'throne', 0.7332440614700317),
 (u'kings', 0.7032474875450134),
 (u'crowned', 0.7003846764564514),
 (u'monarch', 0.6924914717674255),
 (u'prince', 0.6895323395729065),
 (u'eochaid', 0.6760289669036865),
 (u'son', 0.6715421676635742),
 (u'reigned', 0.6627429127693176),
 (u'vii', 0.6580543518066406),
 (u'reign', 0.6530766487121582)]
```

WordRank

```
In [26]: model.most_similar('king')
Out[26]:
[(u'eochaid', 0.8079677820205688),
 (u'canute', 0.792285144329071),
 (u'mormaer', 0.7795065641403198),
 (u'capet', 0.7787382006645203),
 (u'bouillon', 0.7762842178344727),
 (u'alpin', 0.775291204452514б),
 (u'godwinson', 0.7732000946998596)
 (u'gundahar', 0.771564781665802),
 (u'conradin', 0.7687084078788757),
 (u'chatillon', 0.7666929960250854)
```

Word2Vec

```
In [32]: model.most_similar('king')
Out[32]:
[(u'thrones', 0.7961102724075317),
 (u'son', 0.7955597639083862),
 (u'godred', 0.7742120027542114),
 (u'therion', 0.7590411901473999),
 (u'pretender', 0.7583349347114563),
 (u'prince', 0.7535479664802551),
 (u'thron', 0.7528668642044067),
 (u'throne', 0.7479698657989502),
 (u'godfred', 0.7392275333404541),
 (u'pretended', 0.7372602820396423)]
```

FastText

Comparisons to Word2Vec and FastText with TensorBoard visualizations.

Parul Sethi's bio:

Undergrad student of Maths and IT at CIC, University of Delhi. RaRe Incubator Student. GSoC'17 with Gensim

Text Summarization in Pyth[...] Extractive vs. Abstractive techniques revisited

PRANAY, AMAN AND AAYUSH / 2017-04-05 /
GENSIM, STUDENT INCUBATOR, SUMMARIZATION

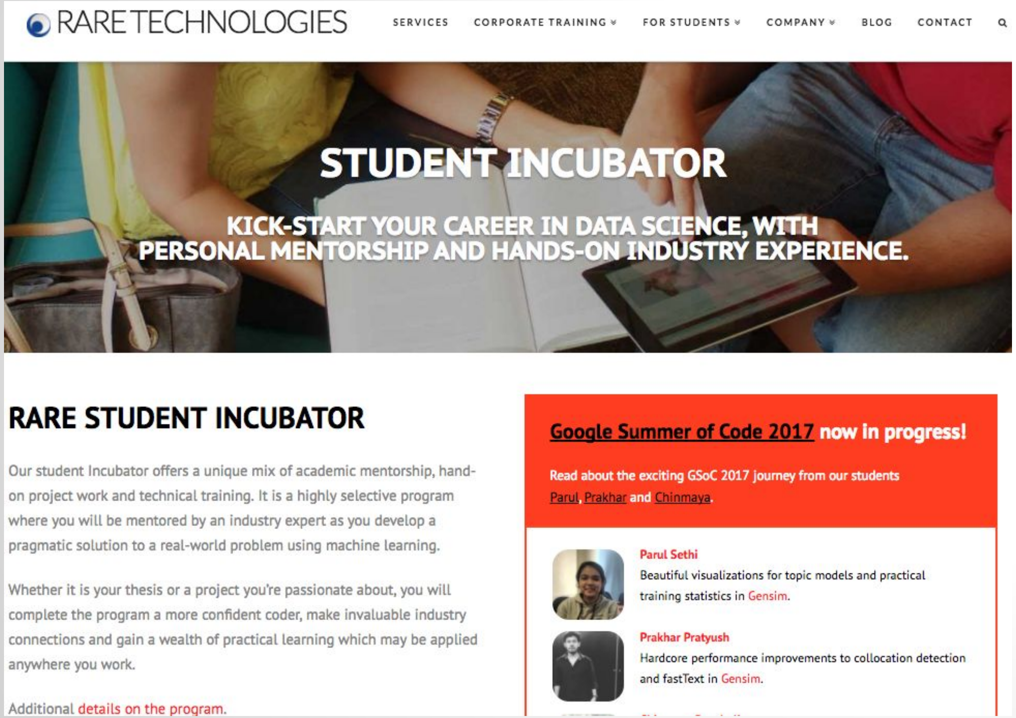
This blog is a gentle introduction to text summarization and can serve as a practical summary of the current landscape. It describes how we, a team of three students in the RaRe Incubator programme, have experimented with existing algorithms and Python tools in this domain.

Pranay, Aman and Aayush's

Ownership & Sustainability Bridge #2: Financial support

- Support talented students: BSc, MSc, PhD
- 1-on-1 mentoring, teach ownership by doing:
    - social: group collaboration, task planning
    - tooling: git, SSH, remote work, testing
    - sanity checking, evaluation
    - presentation: blogs, visualizations
- Sponsor hackathons, meetups, conferences
- Support open source, standard implementations
- Organize competitions

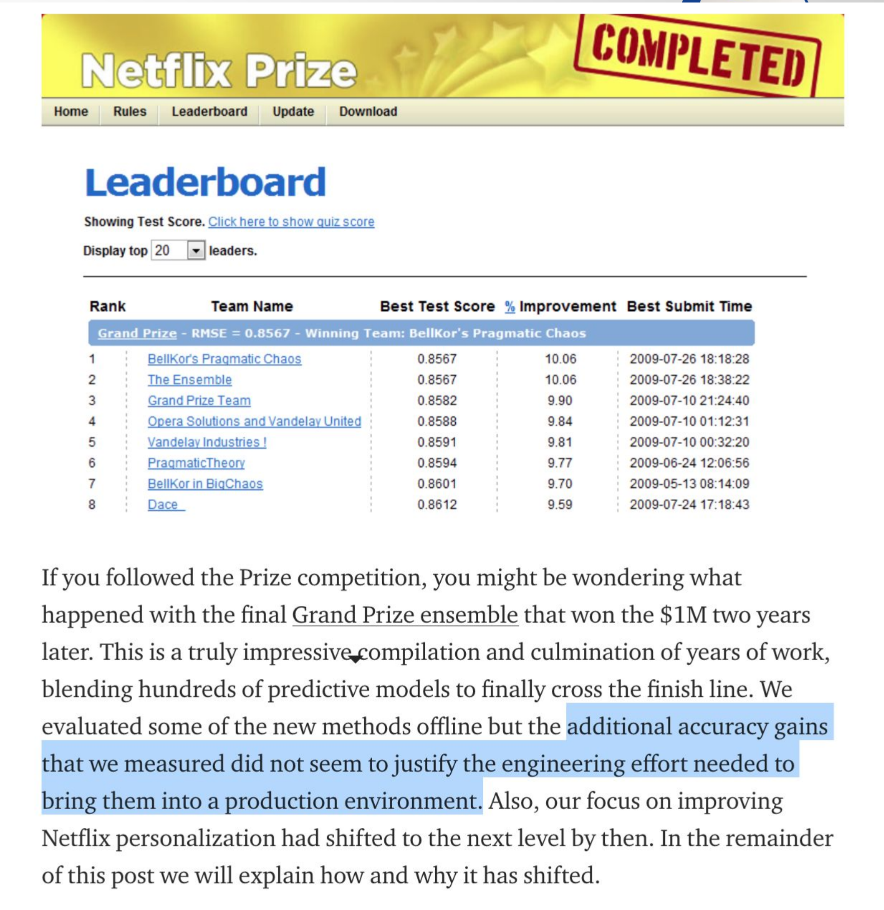
Ownership & sustainability bridge #2:
Financial support



On competitions...

- Good: practical tasks, valuable datasets
- Bad: data hacking, silly winning ensembles, brittle models
  - inevitable: players ± same intelligence as rule makers, but greater in numbers
- Teaches quality (maybe), but still not ownership

Real competition heroes = ppl who prepare the tasks and data?



Netflix Prize COMPLETED

Home | Rules | Leaderboard | Update | Download

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43

If you followed the Prize competition, you might be wondering what happened with the final Grand Prize ensemble that won the $1M two years later. This is a truly impressive compilation and culmination of years of work, blending hundreds of predictive models to finally cross the finish line. We evaluated some of the new methods offline but the additional accuracy gains that we measured did not seem to justify the engineering effort needed to bring them into a production environment. Also, our focus on improving Netflix personalization had shifted to the next level by then. In the remainder of this post we will explain how and why it has shifted.

Ownership & sustainability bridge #3: Provide entropy

- The world changes constantly
  - What is **worth optimizing**? When is stuff **good enough**?
- Subject Matter Expertise GOLD
- Science needs external validation and feedback to avoid **problem overfit**.
- A well-articulated business problem can launch entire research disciplines.

Liling Tan
@alvations

Following

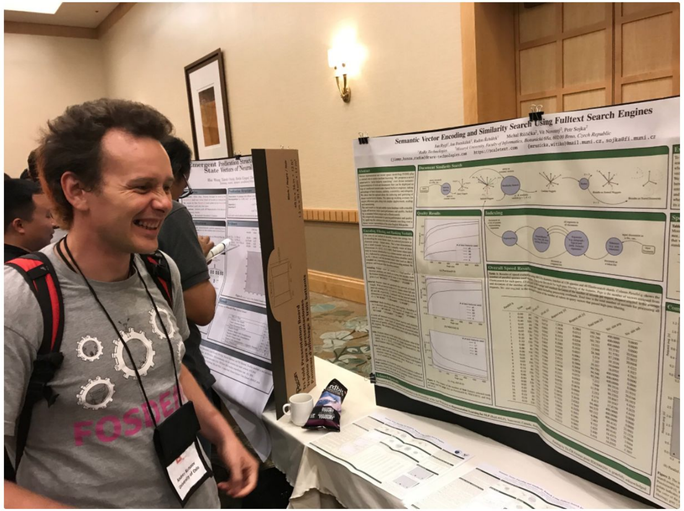
Language identification is viewed as a task as solved as #nlproc. 99.9% means it makes 1000 errors out of 1M and 1M errors out of 1B.

> **Radim Řehůřek** @RadimRehurek
> Beware: the CLD2 algo performs a bit worse on very short texts (compared to langid or langdetect). CLD3 not stable yet. twitter.com/opencpu/status…

1:18 AM - 6 Jun 2017

Ownership & sustainability bridge #3: Provide entropy

- National and EU consortial projects (Horizon2020)
  - Industry to provide data and use-cases
  - Academia to publish research
  - Industry to provide feedback on applicability
- Private research increasingly more important
  - Keep sharing data, infrastructure, tools, know-how



ACL was a blast: lots of amazing people, discussions and suggestions for ScaleText.
Vancouver is quite a way away from our HQ in Prague, but well worth the trip.

Ownership & Sustainability bridge #4: BigCos and GigaLabs

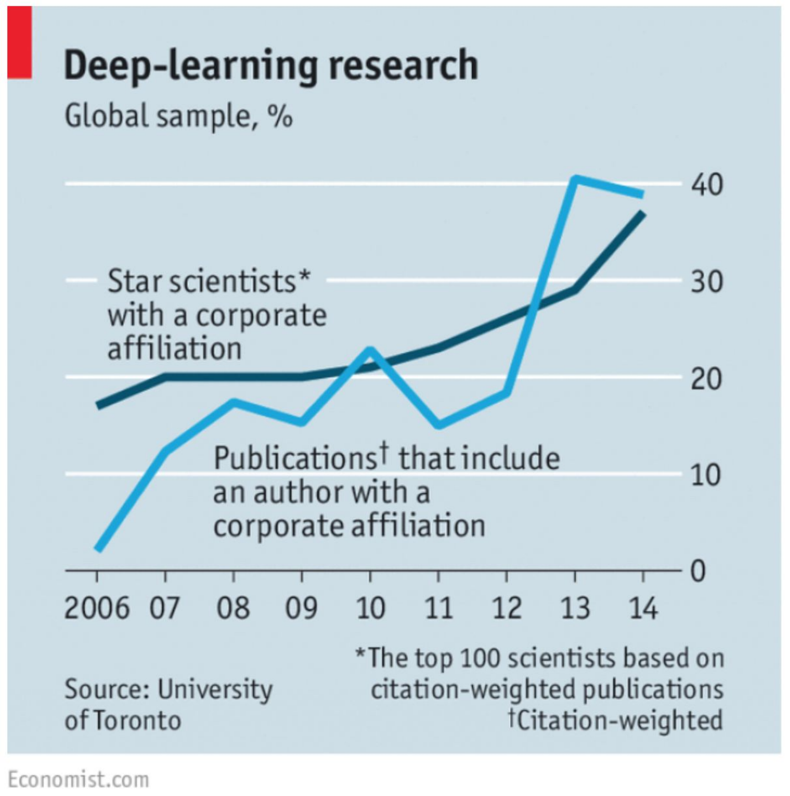
- Lobby for a higher academic impact of non-pub artifacts (SW, tools, repeat studies...).
  - vs the publishing industry racket
- Reduce dependence on an “academic” career
  - Cross-pollinate: open environment, researchers cycle.
  - Helps the SOTA/entropy problem too.
  - Traditional research institutions for Non-BigCos benefit.
- Less focus on ultra-permissive licenses, sets a non-sustainable standard.

Ownership & sustainability bridge #4:
BigCos and GigaLabs

OpenAI

TensorFlow

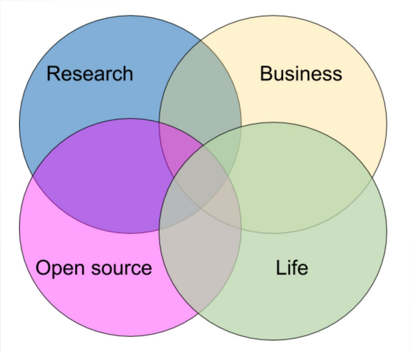
GLUON

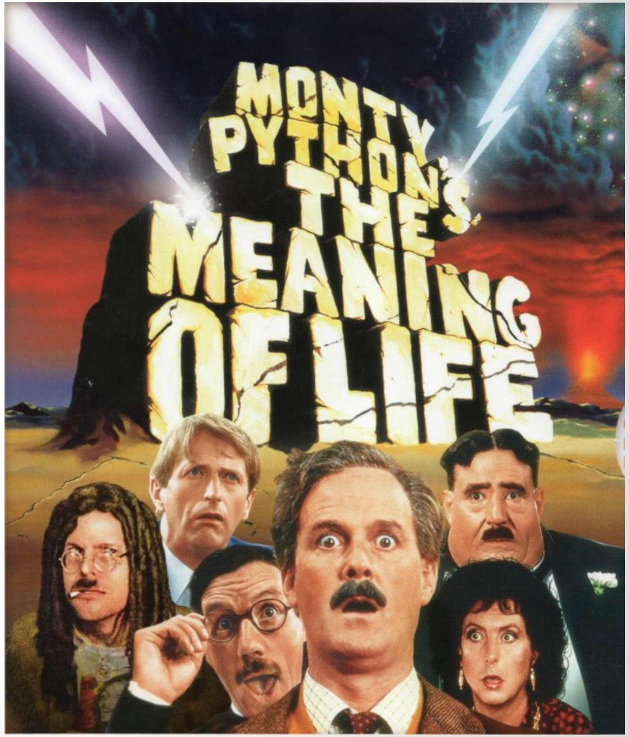


The Bridge of Respect

Pointy haired mng vs *ivory towers* vs *sleazy marketing* vs *clueless engineers* vs *snake oil salesmen* vs *dishonest lawyers*...

Everyone running as hard as they can!





Building bridges: Summary

Academia

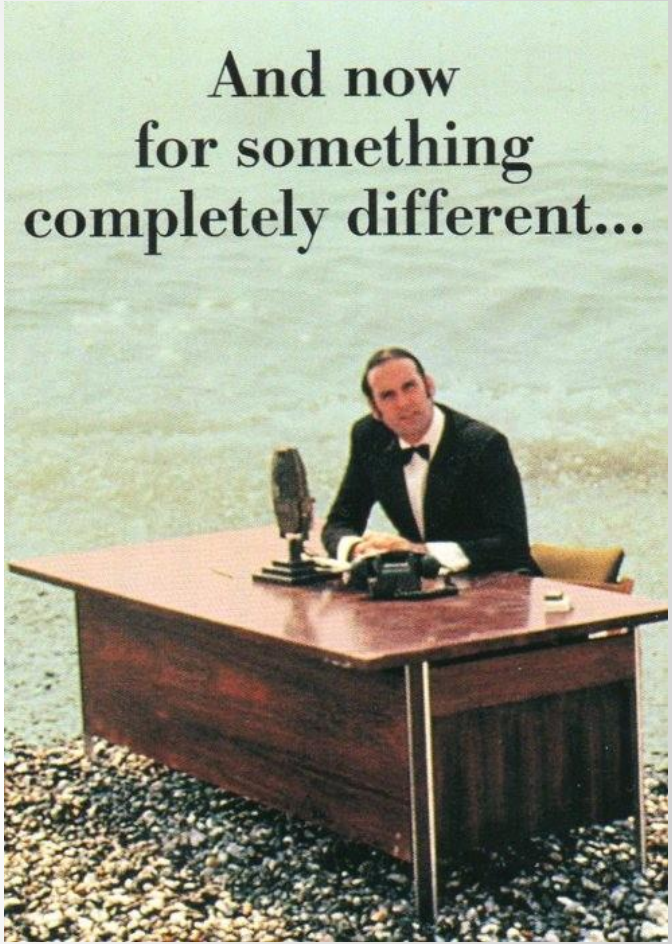
- Realize **unaddressed risk** is nr. 1 rage-factor for companies
- Embrace context, new modalities to present and support results
- Take ownership of results
- Walk before fly

Industry

- Inject entropy, provide utility feedback & data for academic problems
- Actively participate in building skills outside of academic core expertise
- Share resources, sponsor joint events, mentorships, open source
- Lobby for academic incentives of quality & ownership
- Deemphasize SOTA: demand introspection, insights, error analyses

Bonus announcement

Releasing a new open source library: **Bounter**!



Counter from stdlib

```
from collections import Counter

counts = Counter()
counts.update([u'a', 'few', u'words', u'a', u'few', u'times'])  # count item frequencies

print(counts[u'few'])  # query the counts
2
```

A useful class (since Python 2.7):

- count freq distribution of events in logs
- in ML and NLP: building dictionaries, count event co-occurrences, n-grams, collocations, ...

Collocations on EN Wikipedia

Collocation = a group of consecutive words that typically go together:

- Useful to treat as a single unit of information in NLP.
- “New York”, “Olympic Games”, "network license", "Supreme Court" or "elementary school".
- Detect automatically, e.g. Pointwise Mutual Information (PMI)

Challenge: need frequencies of tokens, 2-grams, ...

```python
with smart_open('wikipedia_tokens.txt.gz') as wiki:
    for line in wiki:
        words = line.decode().split()
        bigrams = zip(words, words[1:])
        counter.update(u' '.join(pair) for pair in bigrams)
```

Why Bounter?

Counter / dict needs 31 GB RAM!

- 179,413,989 distinct bigrams out of 1,857,420,106 total.
- + Python’s object overhead.

```
typedef struct {
    PyObject_VAR_HEAD
    long ob_shash;
    int ob_sstate;
    char ob_sval[1];

    /* Invariants:
     *     ob_sval contains space for 'ob_size+1' elements.
     *     ob_sval[ob_size] == 0.
     *     ob_shash is the hash of the string or -1 if not computed yet.
     *     ob_sstate != 0 iff the string object is in stringobject.c's
     *       'interned' dictionary; in this case the two references
     *       from 'interned' to this object are *not counted* in ob_refcnt.
     */
} PyStringObject;
```

Bounter

- “Memory-bounded Counter”.
- Key observation: Exact counts not terribly important (especially in the high-frequency ranges) => approximative algorithms!
- Written in C + Python API ala Counter.

```
from bounter import bounter

counts = bounter(size_mb=1024)  # use at most 1 GB of RAM
counts.update([u'a', 'few', u'words', u'a', u'few', u'times'])  # count item frequencies

print(counts[u'few'])  # query the counts
2
```

Bounter under the hood

Contains 3 algos, progressively more functionality:

1. *cardinality estimation*: HyperLogLog (kBs RAM for billions items)

```
counts = bounter(need_counts=False)
print(counts.cardinality())  # cardinality estimation
print(counts.total())  # efficiently accumulates counts across all items
```

2. + *also individual item counts*: Count-Min Sketch

```
counts = bounter(need_iteration=False, size_mb=200)
print(counts['python'])  # supports asking for counts of individual items
```

3. + *also items()/keys()/iteritems() etc*: optimized hash table

```
counts = bounter(size_mb=200)  # default version, unless you specify need_items or need_counts
counts.update(['a', 'b', 'c', 'a', 'b'])
print(list(counts)) # iterator returns keys, just like Counter
print(list(counts.iteritems()))  # supports iterating over key-count pairs, etc.
```

Benefits of Bounter

We compared the set of collocations extracted from Counter (exact counts, needs lots of memory) vs Bounter (approximate counts, bounded memory) and present the precision and recall here:

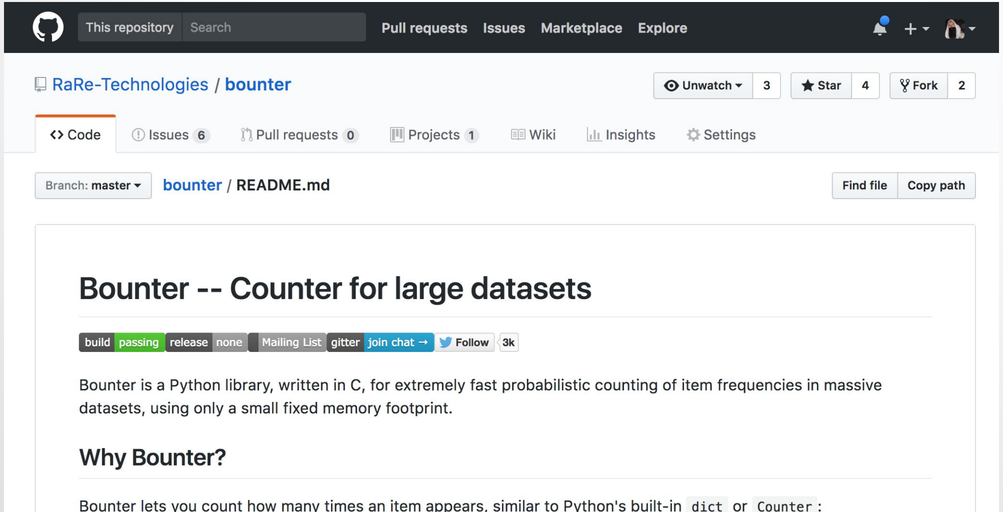
Algorithm	Time to build	Memory	Precision	Recall	F1 score
`Counter` (built-in)	32m 26s	31 GB	100%	100%	100%
`bounter(size_mb=128, need_iteration=False, log_counting=8)`	19m 53s	**128 MB**	95.02%	97.10%	96.04%
`bounter(size_mb=1024)`	17m 54s	1 GB	100%	99.27%	99.64%
`bounter(size_mb=1024, need_iteration=False)`	19m 58s	1 GB	0.9964%	100%	99.82%
`bounter(size_mb=1024, need_iteration=False, log_counting=1024)`	20m 05s	1 GB	**100%**	**100%**	**100%**
`bounter(size_mb=1024, need_iteration=False, log_counting=8)`	19m 59s	1 GB	97.45%	97.45%	97.45%
`bounter(size_mb=4096)`	**16m 21s**	4 GB	100%	100%	100%
`bounter(size_mb=4096, need_iteration=False)`	20m 14s	4 GB	100%	100%	100%
`bounter(size_mb=4096, need_iteration=False, log_counting=1024)`	20m 14s	4 GB	100%	99.64%	99.82%

Bounter achieves a perfect F1 score of 100% at 31x less memory (1GB vs 31GB), compared to a built-in `Counter` or `dict`. It is also 61% faster.

Even with just 128 MB (250x less memory), its F1 score is still 96.04%.

Bounter install & support

- MIT license
- get it from:
  - `pip install bounter`
  - https://github.com/RaRe-Technologies/bounter



Thanks!

http://rare-technologies.com

HIRING ML INSTRUCTORS FOR OUR PUBLIC COURSES!

@radimrehurek
@raretechteam
@gensim_py

(open source stickers up front)