

Getting to know python packages with package2vec

Exploring the space of python packages with ML

Devin de Hueck

AI Data Engineering Intern

ddehueck@redhat.com

January 25th, 2020

In this Presentation

1. Introduction
2. Data and Data Collection
3. Building Python Package Representations
4. Evaluating Representations
5. What comes next?

Introduction

The Goal

Create a python package vector space

Find packages with better features, documentation, performance, etc.

Representation Learning

“The success of machine learning algorithms generally depends on data representation” - Bengio et al. 2012

Data and Data Collection



[Help](#) [Donate](#) [Log in](#) [Register](#)

Find, install and publish Python packages with the Python Package Index

Search projects



Or [browse projects](#)

214,362 projects

1,637,107 releases

2,470,072 files

398,100 users

What Makes up a Python Package?

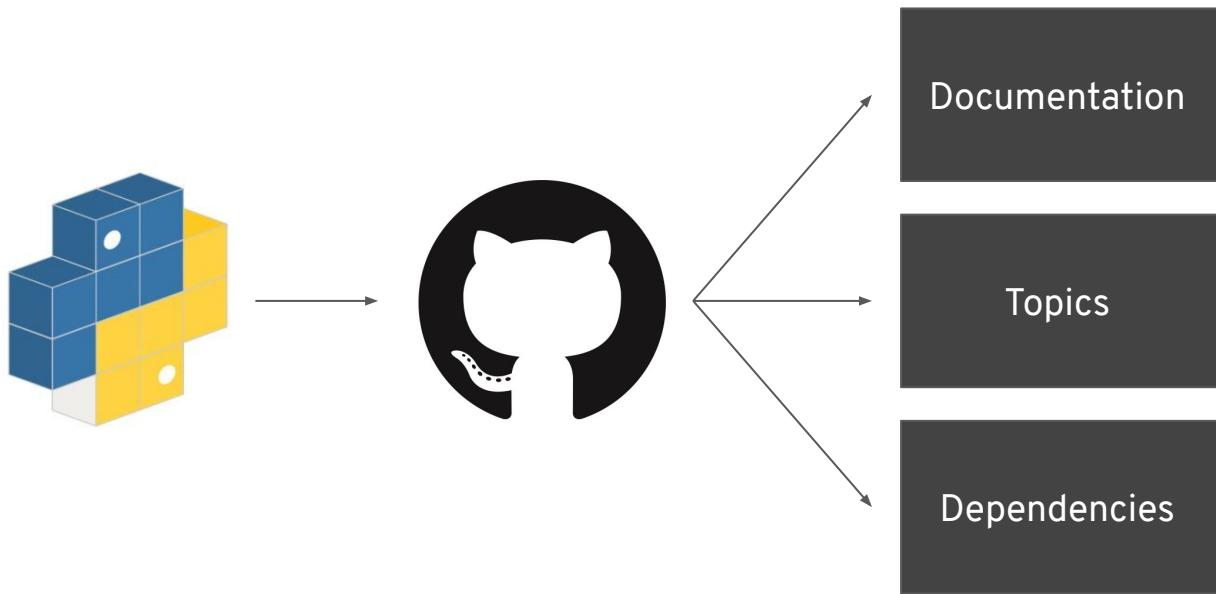
Documentation

Dependencies

The Code

The Community

Collecting the Data



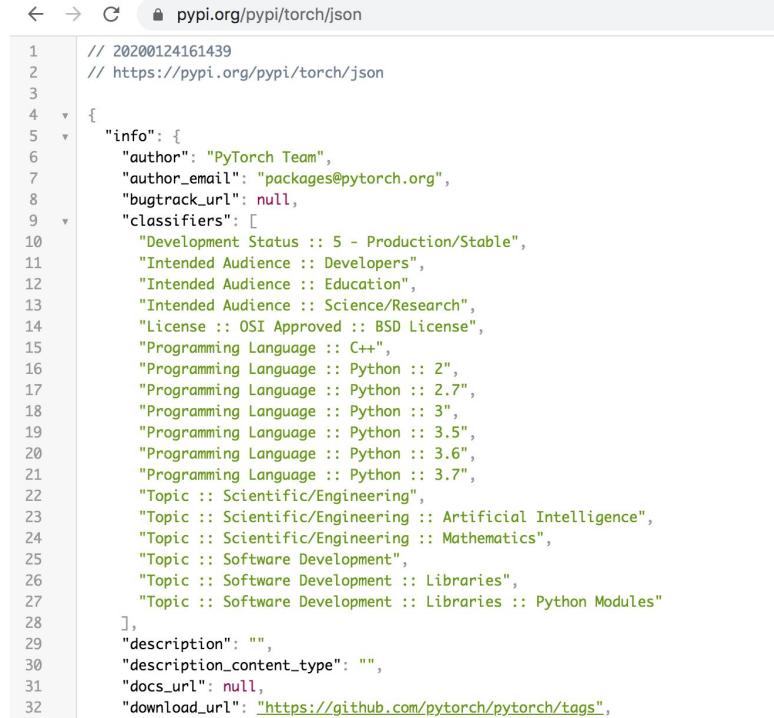
Collecting the Data



The screenshot shows a browser window with the URL `pypi.org/simple/`. The page displays a long list of Python package names, many of which are misspelled or contain errors. The packages listed include:

- 0 0- -0 00000a 0.0.1 007 00print lol 00SMALINUX 0121 01changer 01d61084-d29e-11e9-96d1-7c
- utils 0-orchestrator 0wdg9nbmpm 0x 0x01-autocert-dns-aliyun 0x01-letsencrypt 0x10c-asm 0x-contract-addresses 0x-
- 0x-web3 1 100bot 101703214-assign1-UCS633 101703301-Project1-TOPSIS 101703373-topsis 101703573-Topsis-pk
- configclasses 12factor-vault 131228 pytest 1 1337 153957-theme 15five-django-ajax-selects 15five-snowplow-tracker
- 1ee 1lever-utils 1nester 1OS 1pass 1st 1to001 2 2008WebCrawler 2013007_pyh 20191004 2020 2048 2112 21cmFAST
- 2lazy2rest 2mp3 2mp4 2or3 2to3 2wf90-assignment 3 3-1 311devs peewee 36ban commons 36-chambers 36ke.py 38
- 3Dfunctiongrapher 3d-wallet-generator 3Edit 3gpp-citations 3lwg 3scale-api 3t 3to2 3to2_py3k 3xsd 40wt-common-ta
- XML 5 51degrees-mobile-detector 51degrees-mobile-detector-lite-pattern-wrapper 51degrees-mobile-detector-trie-wra
- 51PubModules 51pub pymodules 5minute 5o4dre15mk 650-auto-comp_jaewon 652ga 69 6D657461666C6F77 6du.tv
- connector 88rest 908dist 91act-platform 91downloader 99d4aa80-d846-424f-873b-a02c7215fc54 a a00k5pgrtn a10ctl :a2d_diary a2m.itertools a2p2 a2pcej a2svm a2w a2x a38 a3cosmos a3cosmos-gas-evolution A3MIO a3rt-sdk-py A3SI
- a4t-terms-and-conditions a569 a5dev a8 a8ctl a8e a99 aa aa2atom aaa aaa103439 aaa2.1.1 Aaaaaaaaaaaaaaaaaaaaaa-aaaaa
- aafootball aa airtable aaa-jiang aaalong aaapi aaapp_nester aaargh aab aabbtree aabc aabc_nester aaboyles aacgmv2
- database-connection Aai aa-intercom aakbar aalam-common aalto-boss aaltopoiju aam aam-api aa-mengjianing aamno
- aapipackage aapns aaps aapt a.arabaci aarc-g002-entitlement aarchimate aarddict aardtools aardvark aardvark-py aarg
- notificationhub aasalert aa-sbst aascraw aasemble aasemble.deployment aashpdf aashupdf aasms aas-timeseries aa-strij
- aaypyutil aa_zwb ab ab2cb aba abacus abacusevents abacusSoftware abacus-tpot ab-addnm abadge abadon-sdk abager
- abaparser abaqus2dyna Abaqus-RunINPFiles abathur ABBA Abbas ab-ble-gateway-sdk-python abb-pro33-ardexa abbi
- abc_algorithm abc-analysis abc-classroom abcd abcde abc-delegation abcdmini ABCD-ML abce abcEconomics abc-gr
- abDB abdbeam abduct abdulpdf Abe abed Abel abel-airflow abelian abellin abem abe-mocks abenity abeona abepdf ab
- optimizers abhilash99 Abhilash-optimizers abhinavPY abhipdf abhiwin_package1 abhorrentTestPackage abi abi2doc a
- abiosgaming.py abipy abiquo-api abito abi.tools.uigenerator A-Bit-Racey Abjad abjad-ext-book abjad-ext-cli abjad-ext
- abl.errorreporter abl.jquery abl.jquery.plugins.form abl.jquery.ui ablk ablog ablog_api ablog_cli abl.robot abl.util abl.v
- aboardly abobo abode abodepy abofly abo-generator aboki aboleth abondance abook abopt abo-s-pysync abot abotest a

Collecting the Data



A screenshot of a web browser window displaying the PyPI JSON endpoint for the torch package. The URL in the address bar is `pypi.org/pypi/torch/json`. The page content is a JSON object representing the package information. The JSON structure includes fields like `info`, `description`, and `download_url`, with many nested values such as classifier names and developer details.

```
// 20200124161439
// https://pypi.org/pypi/torch/json

{
    "info": {
        "author": "PyTorch Team",
        "author_email": "packages@pytorch.org",
        "bugtrack_url": null,
        "classifiers": [
            "Development Status :: 5 - Production/Stable",
            "Intended Audience :: Developers",
            "Intended Audience :: Education",
            "Intended Audience :: Science/Research",
            "License :: OSI Approved :: BSD License",
            "Programming Language :: C++",
            "Programming Language :: Python :: 2",
            "Programming Language :: Python :: 2.7",
            "Programming Language :: Python :: 3",
            "Programming Language :: Python :: 3.5",
            "Programming Language :: Python :: 3.6",
            "Programming Language :: Python :: 3.7",
            "Topic :: Scientific/Engineering",
            "Topic :: Scientific/Engineering :: Artificial Intelligence",
            "Topic :: Scientific/Engineering :: Mathematics",
            "Topic :: Software Development",
            "Topic :: Software Development :: Libraries",
            "Topic :: Software Development :: Libraries :: Python Modules"
        ],
        "description": "",
        "description_content_type": "",
        "docs_url": null,
        "download_url": "https://github.com/pytorch/pytorch/tags"
    }
}
```

Collecting the Data

[pytorch / pytorch](#)

Used by 21.1k Watch 1.4k Star 35.6k Fork 8.9k

Code Issues 3,768 Pull requests 1,148 Actions Projects 5 Wiki Security Insights

Tensors and Dynamic neural networks in Python with strong GPU acceleration <https://pytorch.org>

neural-network autograd gpu numpy deep-learning tensor python machine-learning

23,656 commits 2,673 branches 0 packages 31 releases 1,261 contributors View license

Branch: master New pull request Create new file Upload files Find file Clone or download

suo and facebook-github-bot [jit] Fix dict type serialization (#32569) ... Latest commit 8fd3eae 6 hours ago

.circleci .circleci: Only run macos libtorch on master (#32378) 3 days ago

.ctags.d Add a .ctags.d/ toplevel directory (#18827) 10 months ago

.github move AWS ECR gc jobs to circleci (#30996) last month

.jenkins Move pytorch distributed tests to separate folder for contbuild. (#30445) 2 days ago

android Set rpath for JNI library on Mac (#32247) 3 days ago

aten [pytorch][embeddingbag] Parallelize the EmbeddingBag operator (#4049) 12 hours ago

benchmarks Fix typos, via a Levenshtein-type corrector (#31523) 7 days ago

binaries Fix typos, via a Levenshtein-type corrector (#31523) 7 days ago

Collecting the Data

The screenshot shows the PyTorch GitHub repository's README page. It features the PyTorch logo (a red stylized 'P' with a dot) and the word "PyTorch" in large black letters. Below the logo, a section titled "PyTorch is a Python package that provides two high-level features:" lists:

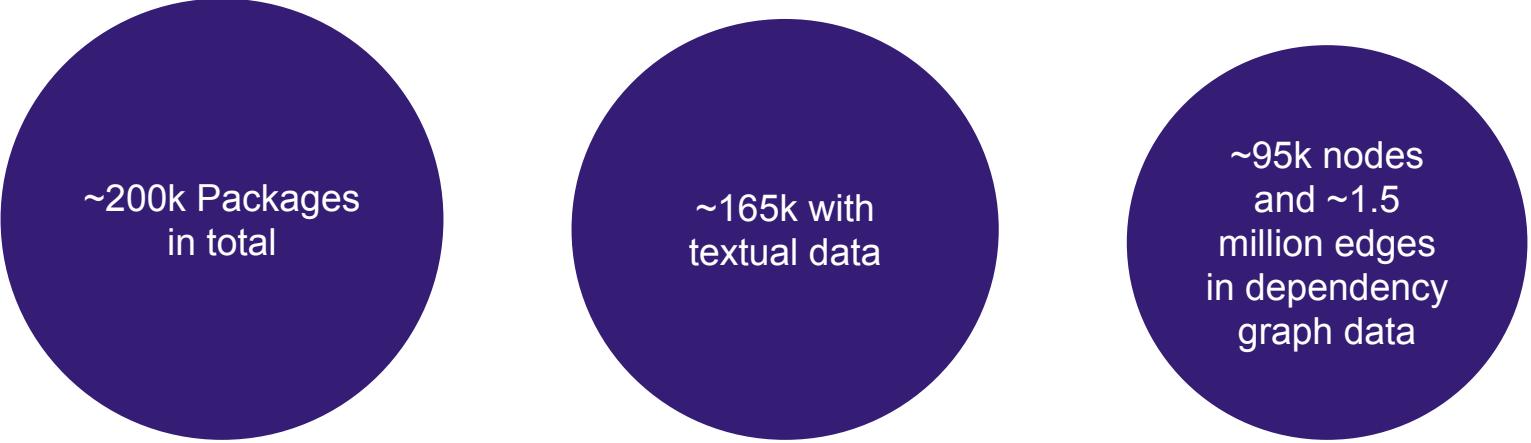
- Tensor computation (like NumPy) with strong GPU acceleration
- Deep neural networks built on a tape-based autograd system

Below this, another section titled "You can reuse your favorite Python packages such as NumPy, SciPy and Cython to extend PyTorch when needed." lists:

- More about PyTorch
- Installation
 - Binaries
 - From Source
 - Docker Image
 - Building the Documentation
 - Previous Versions
- Getting Started
- Communication
- Releases and Contributing
- The Team

The screenshot shows the PyTorch GitHub repository page. At the top, it displays basic statistics: 21.1k used by, 1.4k watch, 35.6k stars, 8.9k forks, 3,768 issues, 1,148 pull requests, and 5 projects. Below this, a sidebar menu includes Pulse, Contributors, Community, Commits, Code frequency, Dependency graph (which is selected), Network, and Forks. The main content area is titled "Dependency graph" and shows a tree view of dependencies defined in requirements.txt. The root node is "PythonCharmers / python-future future". Other nodes include "numpy / numpy", "yaml / pyyaml", "psf / requests", and "pypa / setuptools". A note at the top of the graph area states: "These dependencies are defined in pytorch's manifest files, such as requirements.txt, setup.py, and docs/requirements.txt."

At a Glance



~200k Packages
in total

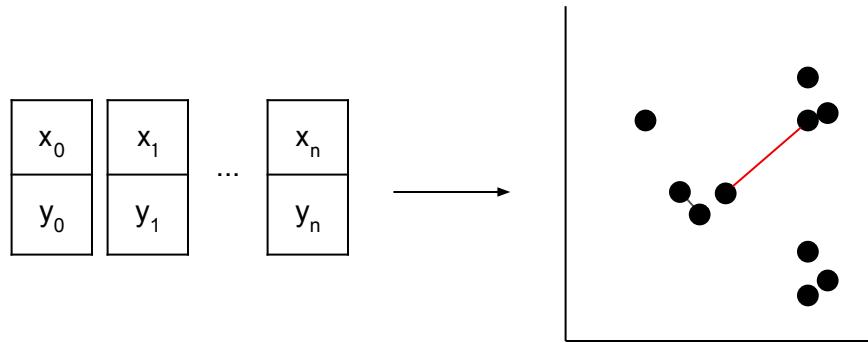
~165k with
textual data

~95k nodes
and ~1.5
million edges
in dependency
graph data

~90k *Have both language and graph data*

Developing Python Package Representations

Vector Spaces for Similarity Metrics



If we can encode python packages as vectors than we have a quantitative similarity metric

Language

TF-IDF

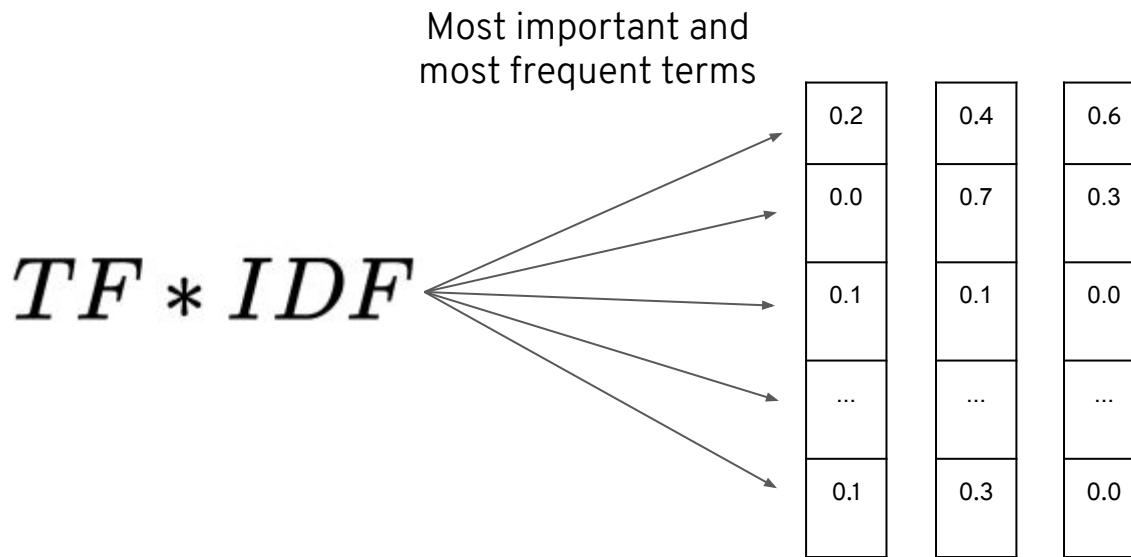
Term Frequency - Inverse Document Frequency

TF-IDF Intuition

$$TF_{i,j} = \frac{\text{Frequency of term } i \text{ in document } j}{\# \text{ Words in document } j}$$

$$IDF_i = \log\left(\frac{\# \text{ of documents}}{\# \text{ documents with term } i}\right)$$

TF-IDF Intuition



TF-IDF - What it tells is

Top mean features across ALL documents

	feature	tfidf
0	image	0.074913
1	target	0.053084
2	django	0.046839
3	install	0.041536
4	api	0.040747
5	alt	0.040086
6	documentation	0.033638
7	add	0.032782
8	code	0.030553
9	module	0.027012
10	file	0.026988
11	pypi	0.025442
12	interface	0.024354
13	package	0.024143
14	license	0.024042
15	plugin	0.023913
16	travis	0.023182
17	datum	0.022962
18	run	0.022431
19	github	0.021816
20	test	0.020548
21	example	0.019522
22	library	0.019413
23	pip	0.017485
24	oca	0.017446

Saving Embeddings...

Saved!

In [7]:

```
1 feature_names = vectorizer.get_feature_names()
2 print(feature_names)

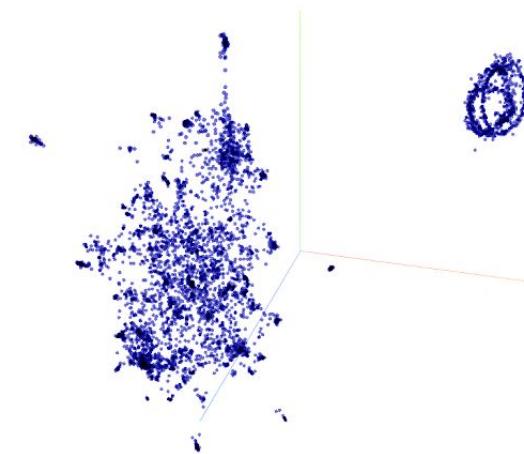
['00', '10', '2017', '2018', 'access', 'account', 'add', 'address', 'allow', 'alt', 'api', 'app', 'application', 'argument', 'attribute', 'author', 'automatically', 'available', 'badge', 'base', 'bash', 'bin', 'block', 'branch', 'bu g', 'build', 'cache', 'case', 'change', 'check', 'class', 'client', 'clone', 'code', 'column', 'command', 'config', 'configuration', 'configure', 'connection', 'contain', 'content', 'context', 'copy', 'coverage', 'create', 'current', 'custom', 'data', 'database', 'date', 'datum', 'def', 'default', 'define', 'delete', 'dependency', 'description', 'de v', 'development', 'different', 'directory', 'display', 'django', 'docker', 'docs', 'document', 'documentation', 'down load', 'easy', 'email', 'enable', 'end', 'environment', 'error', 'event', 'example', 'execute', 'exist', 'extensi on', 'false', 'feature', 'field', 'file', 'filter', 'fix', 'folder', 'follow', 'foo', 'form', 'format', 'function', 'g enerate', 'git', 'github', 'group', 'help', 'host', 'html', 'http', 'https', 'image', 'implement', 'implementation', 'import', 'include', 'index', 'info', 'information', 'input', 'instal', 'install', 'installation', 'instance', 'insta ad', 'interface', 'issue', 'item', 'json', 'key', 'language', 'late', 'level', 'library', 'license', 'like', 'line', 'link', 'list', 'load', 'local', 'log', 'look', 'main', 'master', 'match', 'message', 'method', 'mode', 'model', 'mod ule', 'multiple', 'need', 'new', 'node', 'note', 'number', 'object', 'odoo', 'open', 'option', 'optional', 'order', 'output', 'package', 'page', 'parameter', 'pass', 'password', 'path', 'pip', 'pone', 'plugin', 'png', 'point', 'por t', 'post', 'print', 'process', 'program', 'project', 'provide', 'pypi', 'python3', 'query', 'read', 'reference', 're lease', 'remove', 'report', 'repository', 'request', 'require', 'requirement', 'resource', 'response', 'result', 'ret urn', 'root', 'run', 'sample', 'save', 'script', 'search', 'second', 'section', 'select', 'self', 'send', 'server', 'service', 'set', 'setting', 'setup', 'shell', 'simple', 'single', 'site', 'size', 'software', 'source', 'specific', 'specify', 'standard', 'start', 'state', 'status', 'step', 'store', 'str', 'string', 'style', 'sudo', 'support', 'sv g', 'table', 'tag', 'target', 'task', 'td', 'template', 'test', 'text', 'time', 'title', 'token', 'tool', 'travis', 'true', 'try', 'txt', 'type', 'update', 'url', 'usage', 'use', 'user', 'value', 'variable', 'version', 'view', 'wan t', 'way', 'web', 'work', 'write']
```



Red Hat

Advantages and Disadvantages of TF-IDF

- *Advantages*
 - Interpretability
 - Ease of computation
- *Disadvantages*
 - How to decide what words across the entire vocabulary of python packages?
 - Not memory efficient
 - Poor similarity metric when sparse



Language

Doc2Vec

The Distributional Hypothesis

Words occurring in similar contexts tend to be semantically similar

Word2Vec Intuition

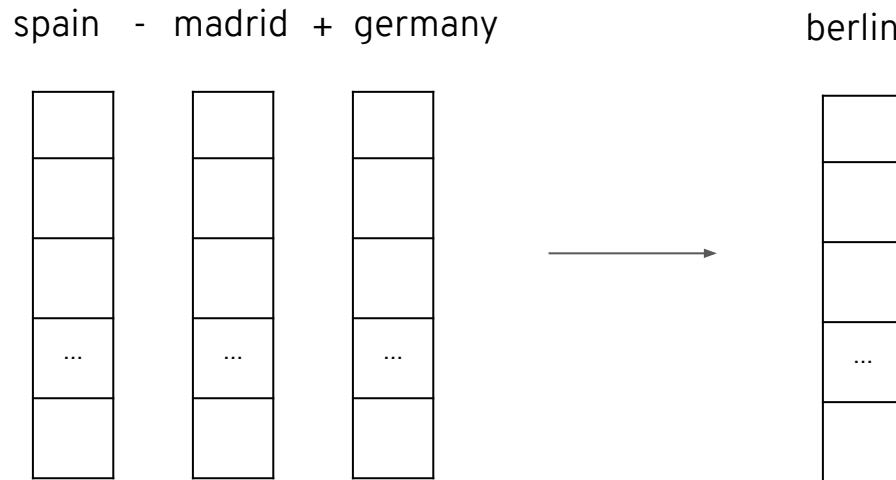
word2vec

“PS!  Thank you for such an
awesome top”

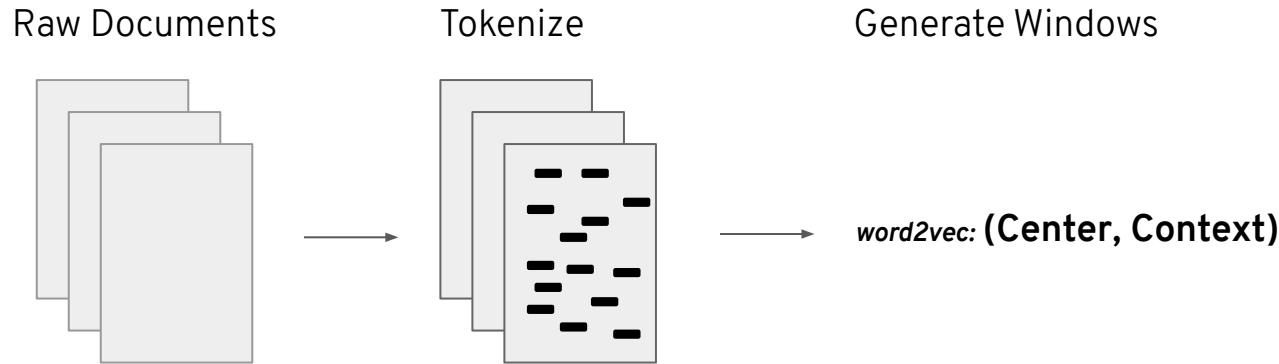
Imagine doing this over billions of sentences.

Interesting Properties of Word2Vec

What is (spain - madrid) + germany?



Training Examples



Word2Vec Negative Sampling Intuition

$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^k \mathbb{E}_{w_i \sim P_n(w)} [\log \sigma(-{v'_{w_i}}^\top v_{w_I})]$$

(Center Word * Context Word) $\longrightarrow \sigma \longrightarrow 1.0/\text{Real}$

(Center Word * Fake Context Word) $\longrightarrow \sigma \longrightarrow -1.0/\text{False}$

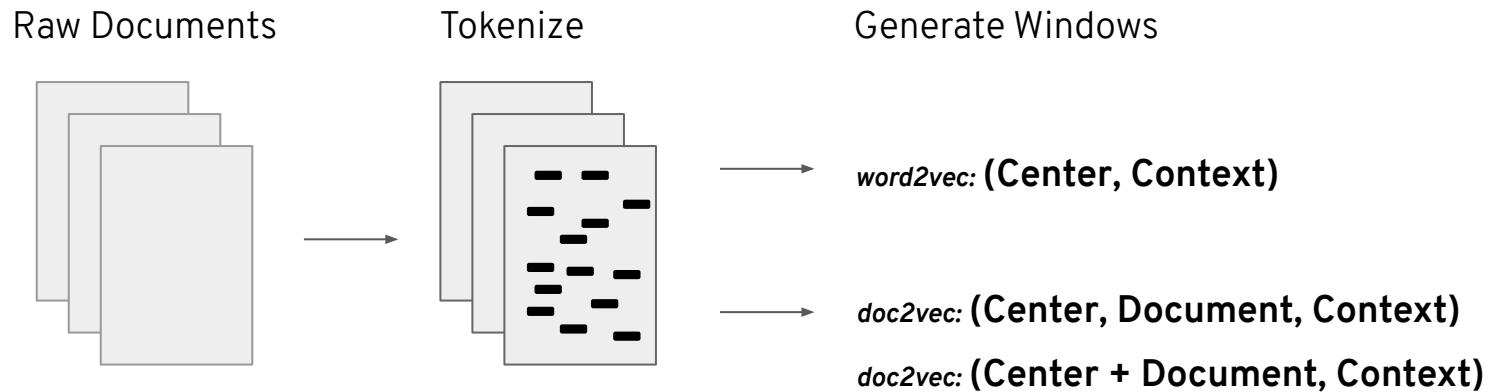
Doc2Vec

The diagram illustrates the Doc2Vec model architecture. At the top right is a green rectangular box labeled "DOC_1732". A green arrow points from this box down to the word "PS!" in a blue box. Another orange arrow points from the word "awesome" in an orange box up to the word "top" in the same orange box. Below these elements is a large, dark gray block of text: "PS! Thank you for such an awesome top".

DOC_1732

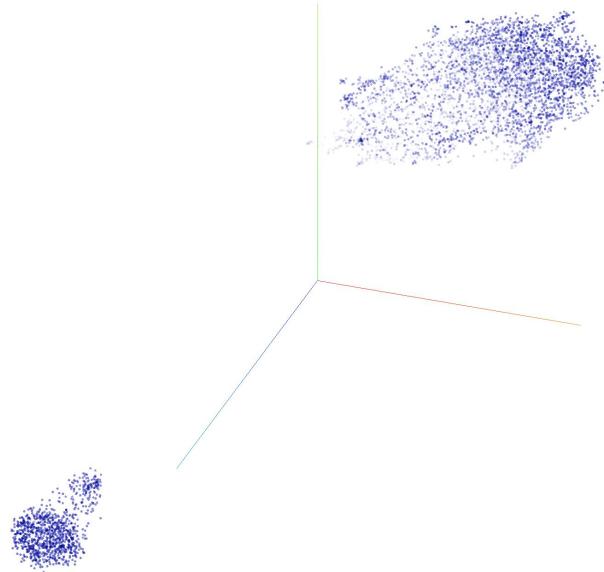
“PS! Thank you for such an
awesome top”

Training Examples



Advantages and Disadvantages of Doc2Vec

- *Advantages*
 - Semantically meaningful distance metric
 - A dense representation let's us use less memory than TF-IDF vectors
- *Disadvantages*
 - A dense representation is not interpretable



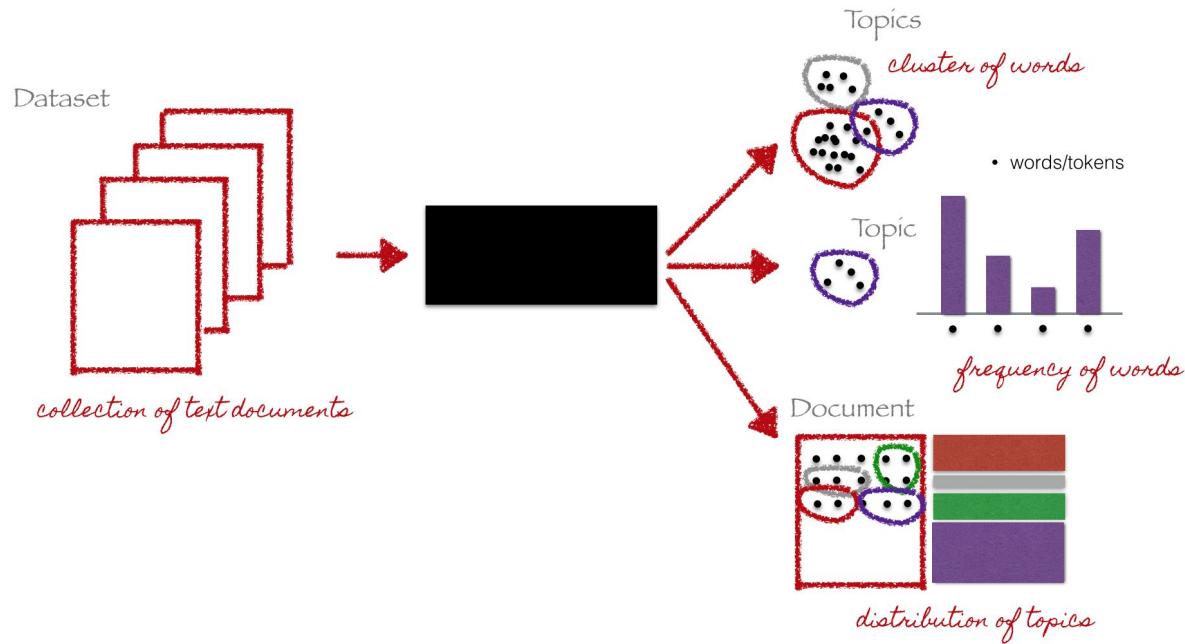
Ash API Demo

Language

ETM

Topic Modeling in Embedding Spaces

Latent Dirichlet Allocation



Latent Dirichlet Allocation

Doesn't work well with a massive vocabulary

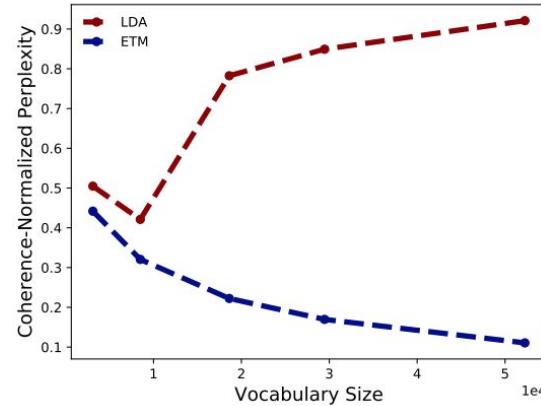


Figure 1. Ratio of the normalized held-out perplexity for document completion and the topic coherence as a function of the vocabulary size for the ETM and LDA. While the performance of LDA deteriorates for large vocabularies, the ETM maintains good performance.



“Games” Topic	“Cloud Computing” Topic	“Machine Learning” Topic	“Web Applications” Topic
Device	Docker	Model	Application
Video	Key	Train	Component
Game	AWS	Learn	Framework
Set	Service	Feature	Provide
Player	Container	Dataset	Core
Play	SSH	Training	Web
Address	Server	Tensorflow	System
Control	Host	Learning	Interface
Audio	Create	Machine	Base

Advantages and Disadvantages of ETM

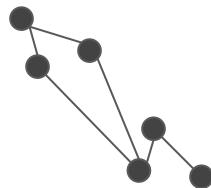
- *Advantages*
 - Learns accurate word embeddings alongside topics
 - Topics allow for some interpretability
 - Beats out LDA in large vocabulary spaces
- *Disadvantages*
 - Doesn't directly learn document embeddings*
 - Have to set number of topics

DeepWalk

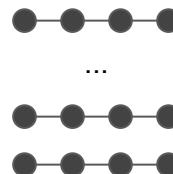
Graphical

Training Examples

Raw Graph



Random Walk

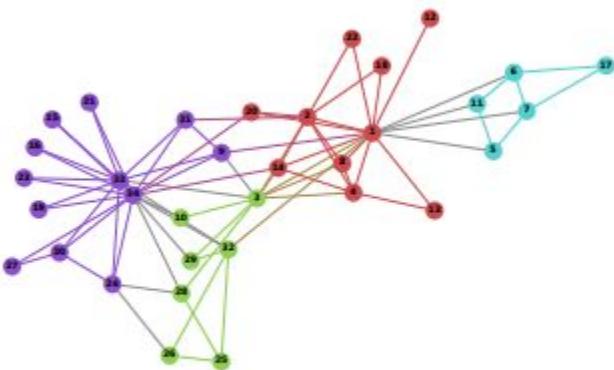


Generate Windows

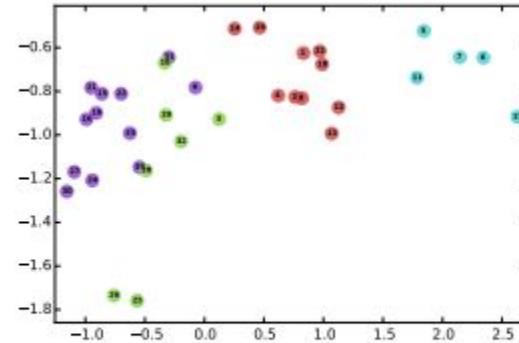
→ (Graph + Package, Context)

A random walk through “the language graph” is a sentence.

Community Graph Structures



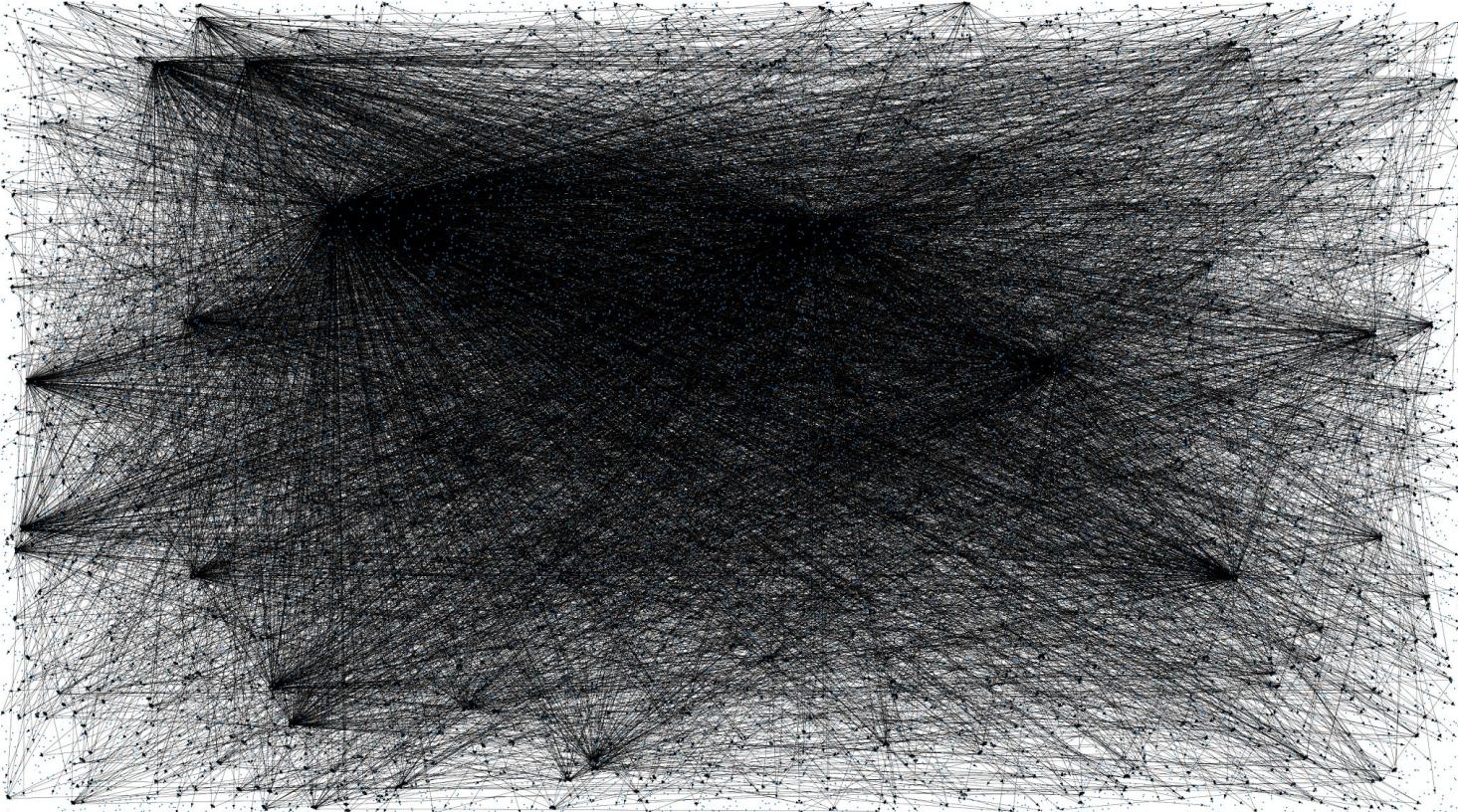
(a) Input: Karate Graph



(b) Output: Representation

Perozzi et al. 2014

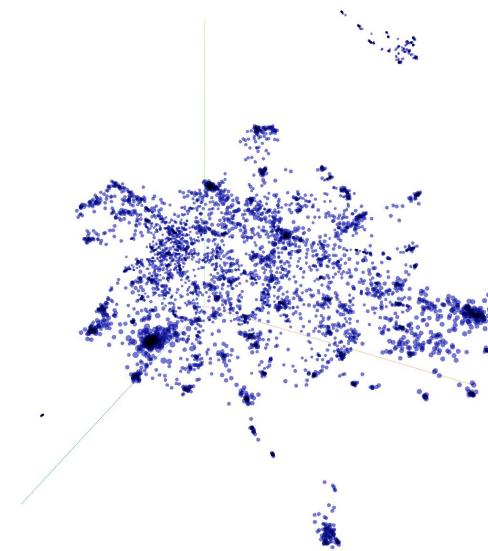
What about PyPI data?



This is 20k packages

Advantages and Disadvantages of DeepWalk

- *Advantages*
 - Opens up a new domain of data!
 - Emphasizes community/social structure
 - **Generalizes language to graph structure**
- *Disadvantages*
 - Not really interpretable



A Joint Approach

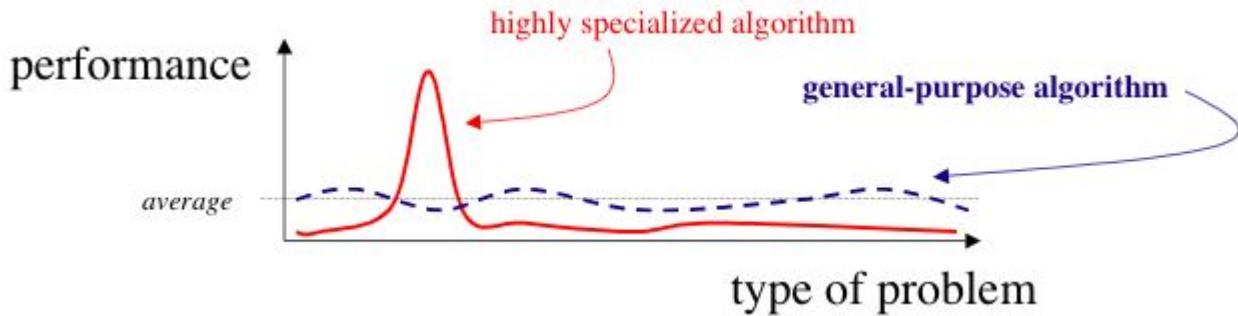
CrossWalk - Walking across data domains

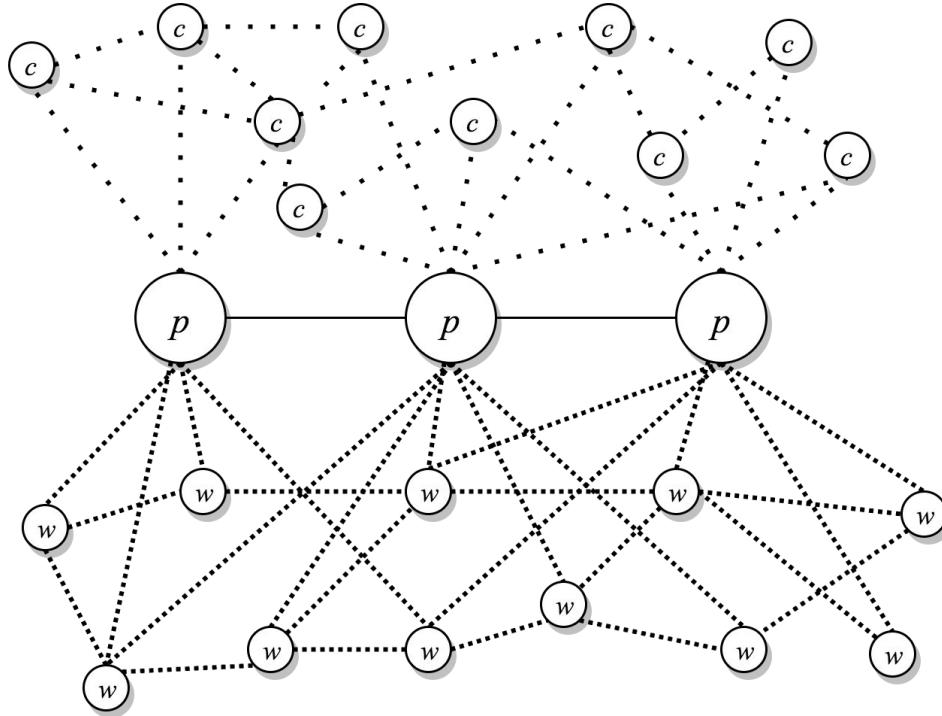
A Joint Approach

Why not use all data sources in one model?

How to interpret our data?

A Joint Approach





p - python package

| - dependency edge

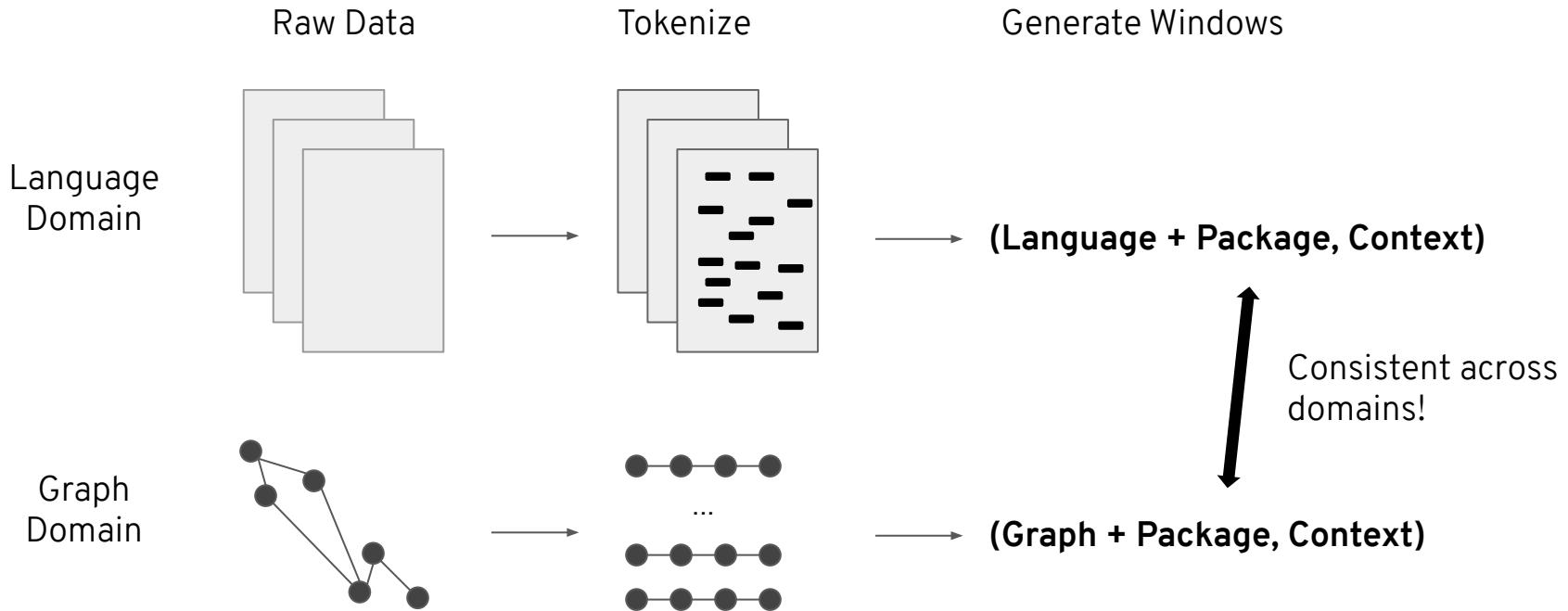
c - code token

: - code edge

w - word token

:: - language edge

Training Examples



Evaluating Python Package Vectors

How to Evaluate?

By the discriminatory ability in the embeddings, i.e. The easier it is to classify the better the representation!

Evaluating with GitHub Topics

The screenshot shows the GitHub repository page for 'pytorch / pytorch'. At the top, there are statistics: 'Used by 21.1k', 'Watch 1.4k', 'Star 35.6k', 'Fork 8.9k'. Below the header, there are tabs for 'Code', 'Issues 3,768', 'Pull requests 1,148', 'Actions', 'Projects 5', 'Wiki', 'Security', and 'Insights'. A banner below the header reads 'Tensors and Dynamic neural networks in Python with strong GPU acceleration' with a link to 'https://pytorch.org'. A red box highlights the 'neural-network' topic in the topic list below. The main content area shows repository metrics: '23,656 commits', '2,673 branches', '0 packages', '31 releases', '1,261 contributors', and a 'View license' button. Below these metrics is a navigation bar with 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. The main feed displays recent commits from various contributors, such as 'suo' and 'facebook-github-bot', with commit details like '#32569' and descriptions like 'Fix dict type serialization'. The commits are timestamped with dates like '6 hours ago', '3 days ago', '10 months ago', and 'last month'.

pytorch / pytorch

Used by 21.1k Watch 1.4k Star 35.6k Fork 8.9k

Code Issues 3,768 Pull requests 1,148 Actions Projects 5 Wiki Security Insights

Tensors and Dynamic neural networks in Python with strong GPU acceleration <https://pytorch.org>

neural-network autograd gpu numpy deep-learning tensor python machine-learning

23,656 commits 2,673 branches 0 packages 31 releases 1,261 contributors View license

Branch: master New pull request Create new file Upload files Find file Clone or download

suo and facebook-github-bot [jit] Fix dict type serialization (#32569) ... Latest commit 8fd3eae 6 hours ago

.circleci .circleci: Only run macos libtorch on master (#32378) 3 days ago

.ctags.d Add a .ctags.d/ toplevel directory (#18827) 10 months ago

.github move AWS ECR gc jobs to circleci (#30996) last month

.jenkins Move pytorch distributed tests to separate folder for contbuild. (#30445) 2 days ago

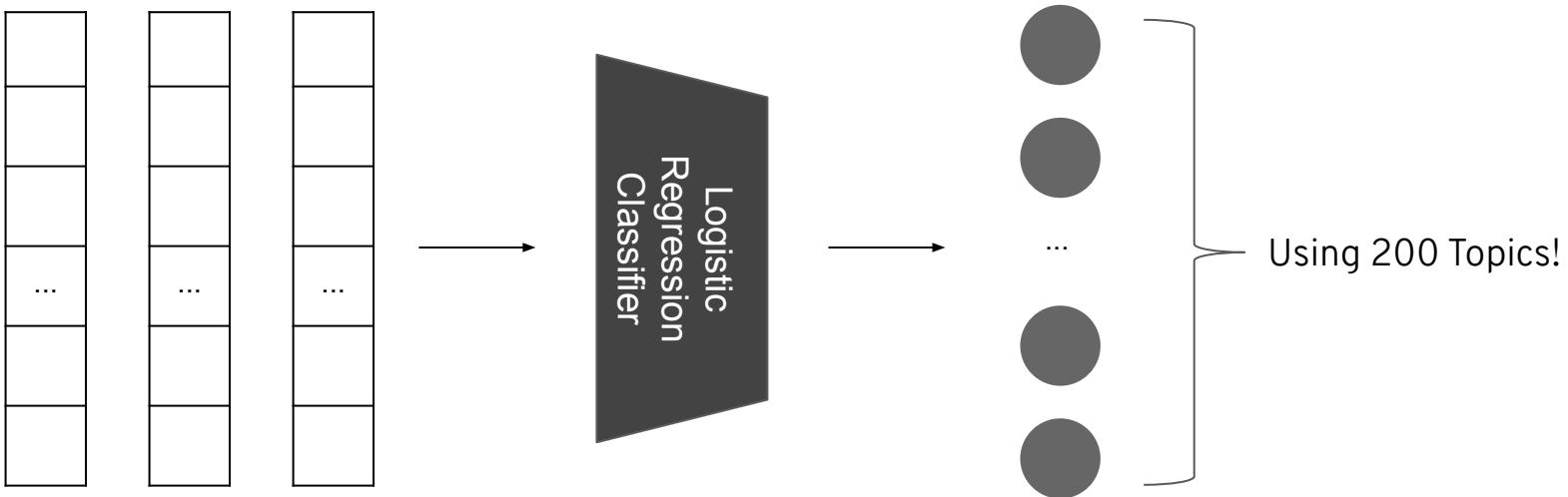
android Set rpath for JNI library on Mac (#32247) 3 days ago

aten [pytorch][embeddingbag] Parallelize the EmbeddingBag operator (#4049) 12 hours ago

benchmarks Fix typos, via a Levenshtein-type corrector (#31523) 7 days ago

binaries Fix typos, via a Levenshtein-type corrector (#31523) 7 days ago

Evaluating with Logistic Regression



Results

Q	S	A	E	C	D	B	T	P	G	O	N	D	Z	M
G	N	P	E	S	A	B	A	T	A	D	B	E	S	A
I	O	I	P	L	T	N	H	G	G	M	I	E	P	C
Z	I	C	Y	A	A	B	P	S	N	I	O	P	T	H
T	T	L	T	R	S	S	C	S	I	C	I	L	E	I
F	A	I	O	S	C	E	O	T	T	R	N	E	N	N
O	Z	E	R	Q	I	C	M	S	S	O	F	A	S	E
S	I	N	C	S	E	U	M	I	E	P	O	R	O	L
O	L	T	H	A	N	R	A	L	T	Y	R	N	R	E
R	A	O	I	D	C	I	N	A	D	T	M	I	F	A
C	U	Y	L	N	E	T	D	M	L	H	A	N	L	R
I	S	N	C	A	Z	Y	L	I	K	O	T	G	O	N
M	I	A	O	P	F	Q	I	N	M	N	I	P	W	I
Y	V	Z	W	S	Z	D	N	I	T	L	C	L	T	N
U	N	L	P	S	J	Z	E	M	N	C	S	O	O	G

MACHINELEARNING	PANDAS
CLI	DATABASE
MICROPYTHON	SECURITY
DEEPLEARNING	VISUALIZATION
AWS	APICLIENT
MINIMALIST	
NLP	
JSON	
DATASCIENCE	
TENSORFLOW	
PYTORCH	
BIOINFORMATICS	
MICROSOFT	
TESTING	
COMMANDLINE	

Results

<i>% Labeled</i>	10%	50%	90%
TF-IDF	30.55%	36.84%	38.40%
Doc2Vec	31.82%	41.03%	43.33%
DeepWalk	34.6%	39.11%	40.53%

In Conclusion

- We built a PyPI dataset - partially labeled
- Tried different embedding techniques and learned along the way
- Developing a model to put it all together
- More to do!

References

1. Bengio, Y., A. Courville, and P. Vincent. "Representation Learning: A Review and New Perspectives. ArXiv e-prints." *arXiv preprint arXiv:1206.5538* (2012).
2. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).
3. Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.
4. Dieng, Adji B., Francisco JR Ruiz, and David M. Blei. "Topic modeling in embedding spaces." *arXiv preprint arXiv:1907.04907* (2019).