Natively

- Wikidata is mostly standard Mediawiki except
- Pages don't store Wikitext, they store JSON blobs

Structure of an Item

- Dictionary
  - {“labels”: language dicitonary
  - “descriptions”: language dicitonary
  - “aliases”: language dicitonary
  - “claims”: list of property and values
  - “sitelinks”: language dicitonary}

Ways to get the data

- Live, from the API, using pywikibot
  - Read / write
- Offline, dumps, using “wda” (WikiData Analytics) parser
  - https://github.com/mkroetzsch/wda
  - Read only
- Linked data, entities and Content negotiation
  - Read only
  -

Content Negotiation

- Path
  - wikidata.org/entity/QID.
    - nt
    - Rdf
    - ttl

Content Negotiaton Example

- https://www.wikidata.org/wiki/Special:EntityData/Q42046.ttl
  - @prefix entity: <http://www.wikidata.org/entity/> .
  - @prefix wikibase: <http://www.wikidata.org/ontology#> .
  - @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
  - @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
  - @prefix schema: <http://schema.org/> .
  - @prefix data: <http://www.wikidata.org/wiki/Special:EntityData/> .
  - @prefix cc: <http://creativecommons.org/ns#> .
  - @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
  -
  - entity:Q42046
  - a wikibase:Item ;
  - rdfs:label "鬣狗科"@zh, "Hienowate"@pl, "Hiena"@eu, "Hyaenidae"@es, "Hiëna"@af, "Dubuk"@ms, "Hiénafélék"@hu, "Fisi"@sw, "Hüäänlased"@et, "হায়েনা"@bn, "Hiena"@sq, "Hyaenidae"@br, "Ύαινα"@el,
  -
  -

Using Pywikibot

- Almost full support of the API
  - New classes in the “core” branch
    - class WikibasePage(Page):
    - class ItemPage(WikibasePage):
    - class PropertyPage(WikibasePage):
      - class Claim(PropertyPage):

Using Pywikibot

- Classic pywikibot pagegenerators work.
  - #make a generator for all the pages with a property
  - en_wikipedia = pywikibot.Site('en', 'wikipedia')
  - wikidata = en_wikipedia.data_repository()
  - property_page = pywikibot.ItemPage(wikidata, 'Property:P21')
  - pages_with_property = property_page.getReferences()

Pywikibot example

- Harvesting infoboxes. In this case book, and dececting similar genres

Using wda

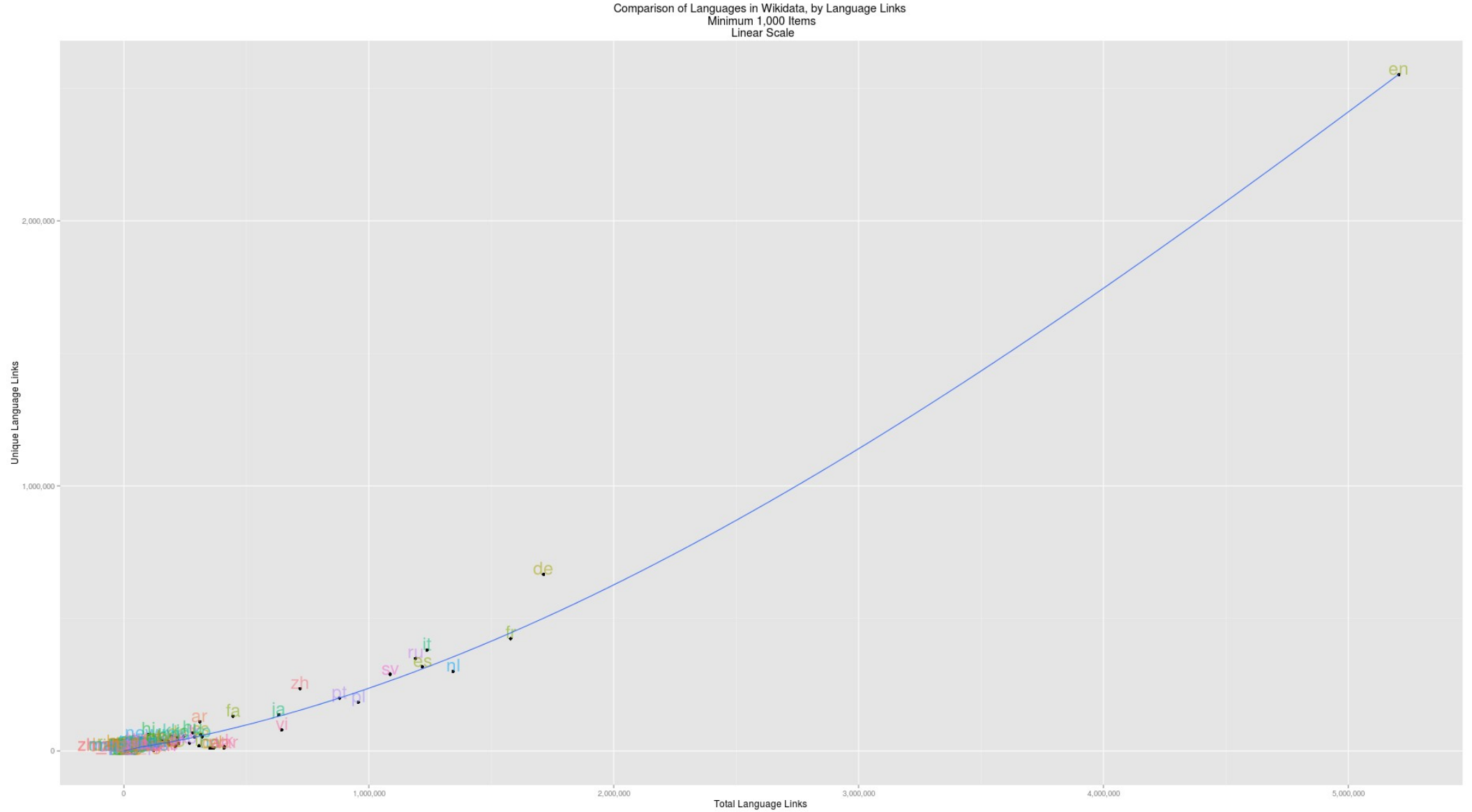
- Downloads dump
  - And then uses nightly incremental dumps
  - About 10GB first download
- Also written in python

Using wda

- After downloading there is a parser that writes a file called kb.txt
  - kb.txt stores plaintext triples
    - one per line
  - 21 link {cdowiki:Ĭng-gáik-làng} .

- Q21 link {trwiki:İngiltere} .
- Q21 link {hewiki:אנגליה} .
- Q21 alias {en:ENG} .
- Q21 alias {min:Inggirih} .
- Q21 alias {sgs:England} .
- Q21 P31 Q1763527 .
- Q21 P47 Q22 .
- Q21 P47 Q25 .
- Q21 P36 Q84 .
- Q21 P85 Q40807 .
- Q21 P85 Q489607 .
- Q21 P85 Q182268 .
- Q21 P17 Q145 .
- Q21 P31 Q3336843 .
- Q21 P41 {Flag of England.svg} .

Wda example


Comparison of Languages in Wikidata, by Language Links
Minimum 1,000 Items
Linear Scale
Unique Language Links
Total Language Links
2,000,000
1,000,000
0
1,000,000
2,000,000
3,000,000
4,000,000
5,000,000
en
de
fr
it
es
nl
ru
sv
pl
pt
zh
ja
vi
fa
ar

Wda example

- Can somebody help with interactive graph visualization?