What is Watson?

Finn Årup Nielsen

DTU Compute
Technical University of Denmark

March 9, 2015

IBM Watson

Note a distinction:

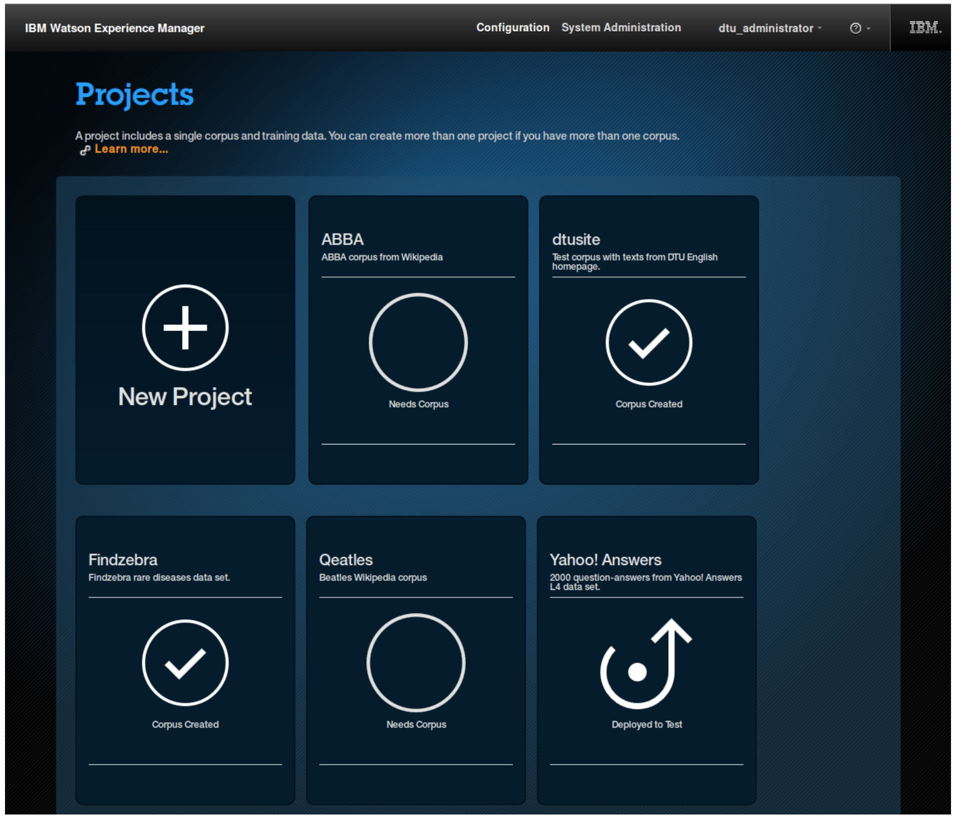
IBM Watson Jeopardy version

Question-answering system with a lot a natural language processing and its own corpus taken from Wikipedia, Wiktionary, DBpedia, YAGO and a lot of other resources (Chu-Carroll et al., 2012) and IBM Watson: Beyond Jeopardy! Q&A. Specialized game control with text-to-speech.

IBM Watson commercial DTU-version

Question-passage retrieval system with a lot of natural language processing where you provide you own corpus as well as question-passages for training Watson on your corpus. Web interface for non-programmer access as well as REST API.

IBM Watson Experience Manager



IBM Watson Experience Manager
Configuration
System Administration
dtu_administrator
IBM.
Projects
A project includes a single corpus and training data. You can create more than one project if you have more than one corpus.
Learn more...
New Project
ABBA
ABBA corpus from Wikipedia
Needs Corpus
dtusite
Test corpus with texts from DTU English homepage.
Corpus Created
Findzebra
Findzebra rare diseases data set.
Corpus Created
Qeatles
Beatles Wikipedia corpus
Needs Corpus
Yahoo! Answers
2000 question-answers from Yahoo! Answers L4 data set.
Deployed to Test

DTU has access to an instance of the IBM Watson question-and-answer part of the IBM cloud services.

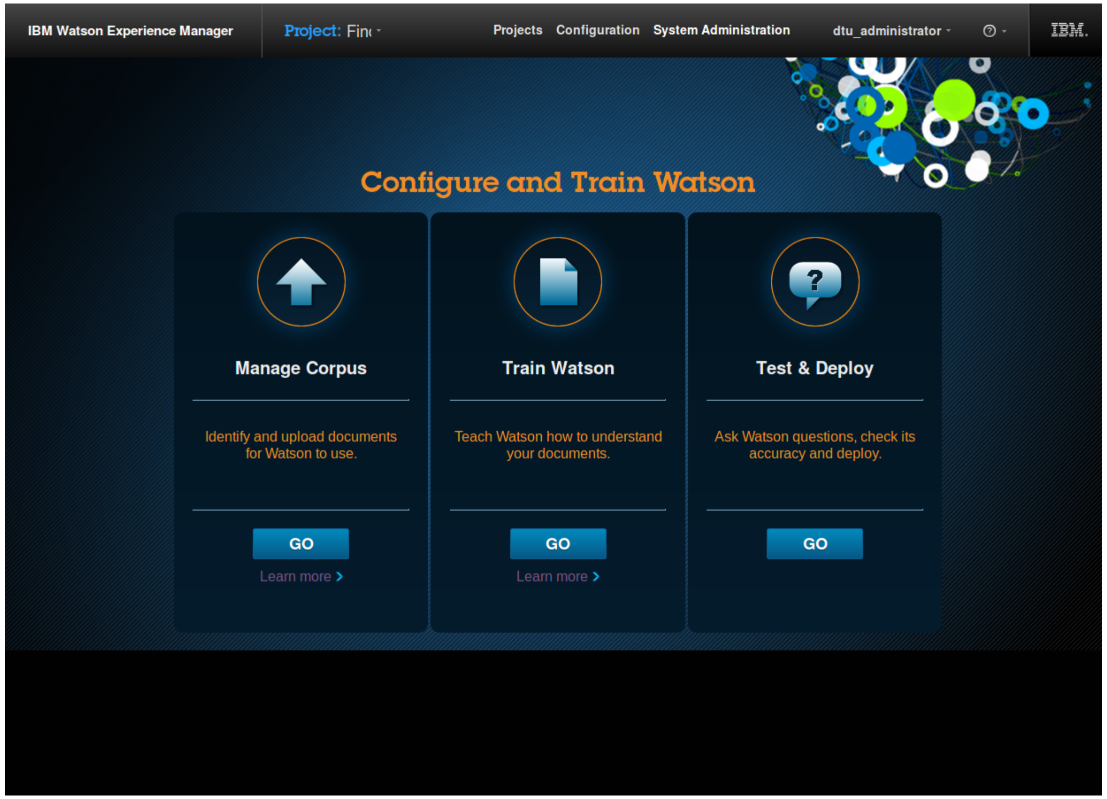
A graphical administration interface is available via the so-called IBM Watson Experience Manager.

Within the DTU instance there are 5 so-called "projects" available.

Each project has an associated corpus and questions linked to passages.

Only one project can be deployed at a time.

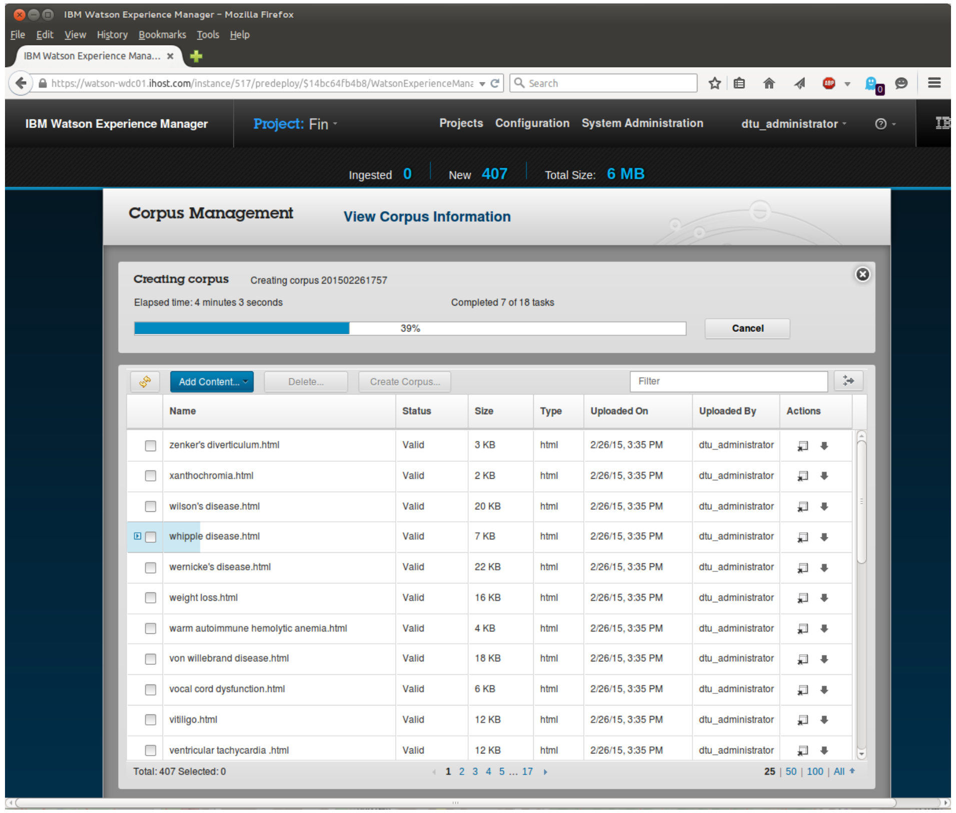
IBM Watson “Configure and train”


IBM Watson Experience Manager
Project: Fin
Projects
Configuration
System Administration
dtu_administrator
IBM.
Configure and Train Watson
Manage Corpus
Identify and upload documents for Watson to use.
GO
Learn more
Train Watson
Teach Watson how to understand your documents.
GO
Learn more
Test & Deploy
Ask Watson questions, check its accuracy and deploy.
GO

Steps in IBM Watson question-and-answer:

1. Upload a corpus.
2. Make questions and linked them to passages (“train”).
3. “Create corpus”.
4. Deploy corpus.
5. Test/use Watson, e.g., from the API.

IBM Watson Corpus manager


IBM Watson Experience Manager – Mozilla Firefox
File Edit View History Bookmarks Tools Help
IBM Watson Experience Mana...
https://watson-wdc01.ihost.com/instance/517/predeploy/$14bc64fb4b8/WatsonExperienceMana
Search
IBM Watson Experience Manager
Project: Fin
Projects
Configuration
System Administration
dtu_administrator
Add Content...
Delete...
Create Corpus...
Filter
Cancel

Ingested 0 | New 407 | Total Size: 6 MB

Corpus Management — View Corpus Information

Creating corpus — Creating corpus 201502261757

Elapsed time: 4 minutes 3 seconds — Completed 7 of 18 tasks

39%

Name	Status	Size	Type	Uploaded On	Uploaded By
zenker's diverticulum.html	Valid	3 KB	html	2/26/15, 3:35 PM	dtu_administrator
xanthochromia.html	Valid	2 KB	html	2/26/15, 3:35 PM	dtu_administrator
wilson's disease.html	Valid	20 KB	html	2/26/15, 3:35 PM	dtu_administrator
whipple disease.html	Valid	7 KB	html	2/26/15, 3:35 PM	dtu_administrator
wernicke's disease.html	Valid	22 KB	html	2/26/15, 3:35 PM	dtu_administrator
weight loss.html	Valid	16 KB	html	2/26/15, 3:35 PM	dtu_administrator
warm autoimmune hemolytic anemia.html	Valid	4 KB	html	2/26/15, 3:35 PM	dtu_administrator
von willebrand disease.html	Valid	18 KB	html	2/26/15, 3:35 PM	dtu_administrator
vocal cord dysfunction.html	Valid	6 KB	html	2/26/15, 3:35 PM	dtu_administrator
vitiligo.html	Valid	12 KB	html	2/26/15, 3:35 PM	dtu_administrator
ventricular tachycardia .html	Valid	12 KB	html	2/26/15, 3:35 PM	dtu_administrator

Total: 407 Selected: 0 — 1 2 3 4 5 ... 17 — 25 | 50 | 100 | All

Data can come from IBM-predefined corpora, e.g., WikiVoyage or it can be uploaded by the administrator of the instance, e.g., one PDF or HTML file or several zipped together.

Watson will segment the documents, based on section heading. In HTML that will be the h1, h2 and h3 tags.

"Creating corpus": Here is (most likely) the natural language processing taking place, where documents and questions are read and processed.

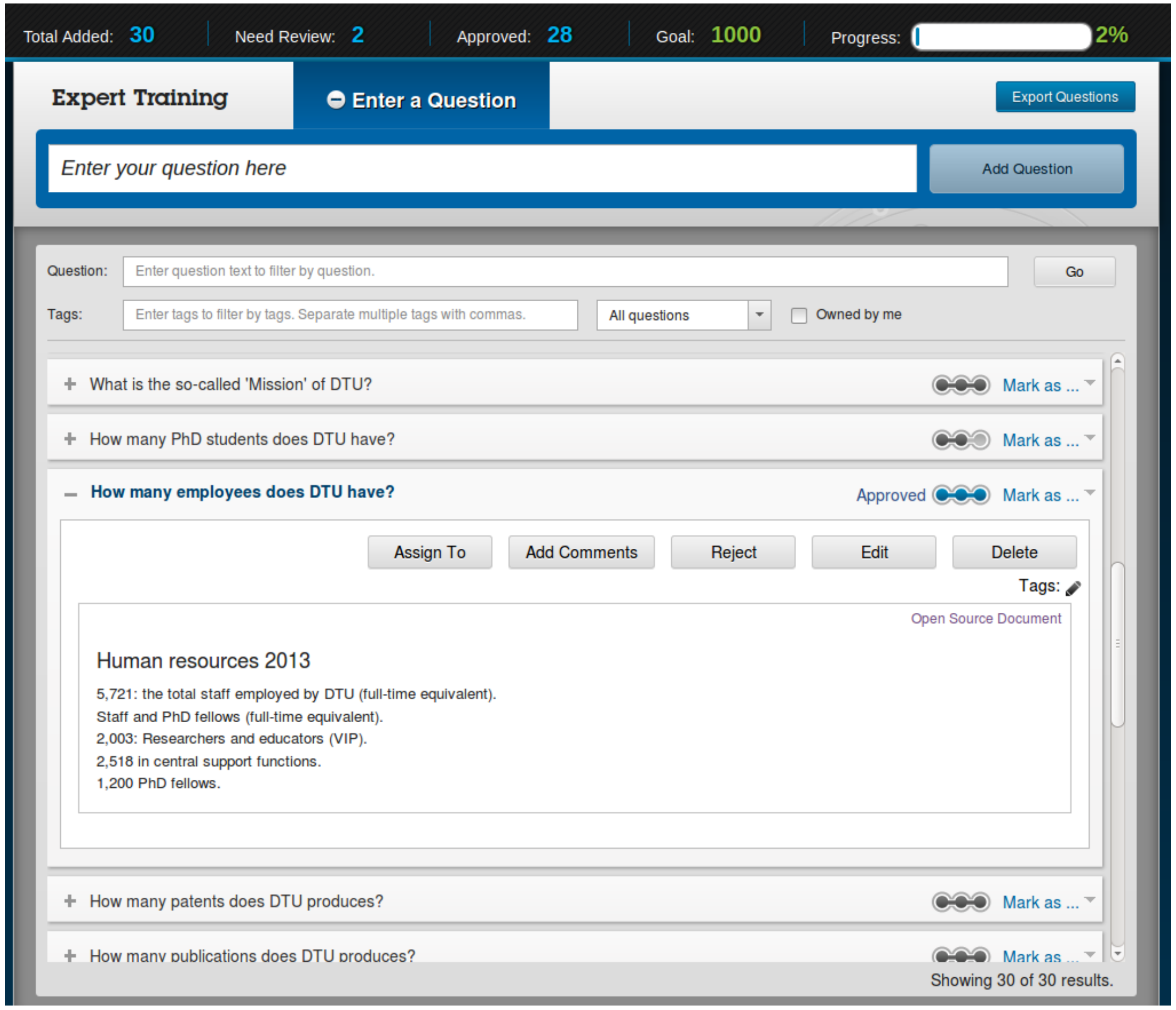
This is a somewhat slow step, e.g., a FindZebra corpus with 407 documents (6MB) takes quarter of an hour to "ingest".

Corpora

Corpora we have tried on IBM Watson so far:

Name	Domain	Documents	Questions
Yahoo! Answers	Open (How. . .)	2,000 (142,627)	2,000 (142,627)
FindZebra	Rare diseases	363	511
DTU English site	DTU	288	(0)
Beatles	Beatles	269	(25)
ABBA	ABBA	126	(0)

IBM Watson training



What is the so-called 'Mission' of DTU?

How many PhD students does DTU have?

How many employees does DTU have?

Human resources 2013

5,721: the total staff employed by DTU (full-time equivalent).
Staff and PhD fellows (full-time equivalent).
2,003: Researchers and educators (VIP).
2,518 in central support functions.
1,200 PhD fellows.

How many patents does DTU produces?

How many publications does DTU produces?

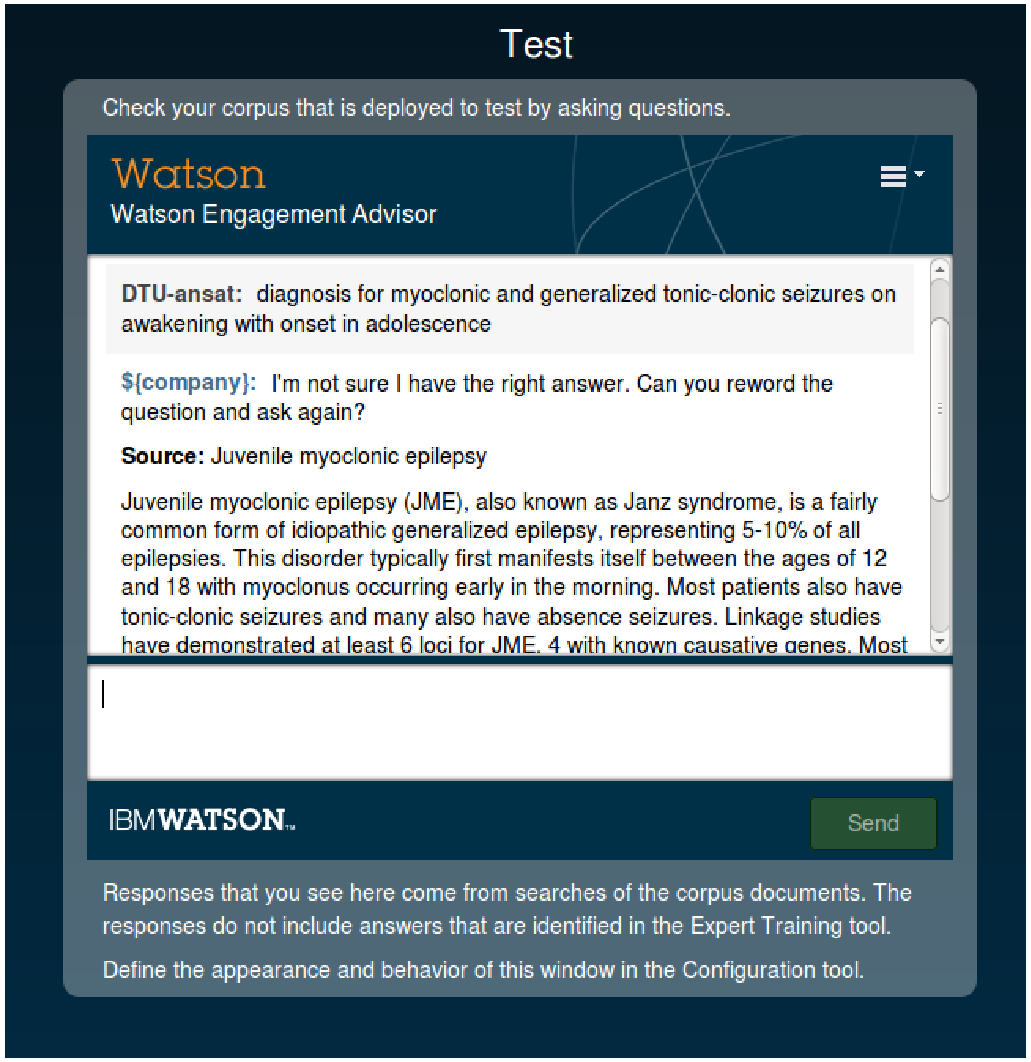
Questions and links to passages can be typed in the web interface.

The interface is relatively easy to use for a non-programmer but requires the questions typed in one by one, — each requiring several steps.

Different user roles for question entry and approval.

Watson would like 1,000 question-passage pairs, — or at least a few hundred.

IBM Watson Test



Test

Check your corpus that is deployed to test by asking questions.

DTU-ansat: diagnosis for myoclonic and generalized tonic-clonic seizures on awakening with onset in adolescence

${company}: I'm not sure I have the right answer. Can you reword the question and ask again?

Source: Juvenile myoclonic epilepsy

Juvenile myoclonic epilepsy (JME), also known as Janz syndrome, is a fairly common form of idiopathic generalized epilepsy, representing 5-10% of all epilepsies. This disorder typically first manifests itself between the ages of 12 and 18 with myoclonus occurring early in the morning. Most patients also have tonic-clonic seizures and many also have absence seizures. Linkage studies have demonstrated at least 6 loci for JME, 4 with known causative genes. Most

Responses that you see here come from searches of the corpus documents. The responses do not include answers that are identified in the Expert Training tool.

Define the appearance and behavior of this window in the Configuration tool.

Once the corpus has been created (new) questions can be typed in the "Test" part of the Web interface.

Here the question is "diagnosis for myoclonic and generalized tonic-clonic seizures on awakening with onset in adolescence" derived from the FindZebra corpus and Doctor's Dilemma.

Watson returns "Juvenile myoclonic epilepsy", — the correct answer.

IBM Watson API

For a programmer it is easier to use the API:

```
>>> import brede.api.watson        # My toolbox with a Watson API
>>> watson = brede.api.watson.Watson()
>>> response = watson.ask(('diagnosis for myoclonic and generalized '
                           'tonic-clonic seizures on awakening with '
                           'onset in adolescence'))
>>> response.evidencelist[0]['title']
u'Juvenile myoclonic epilepsy'
```

My toolbox which includes a Watson module: https://github.com/fnielsen/brede (note the draft branch may be more developed).

For Python there is also pywatson.

IBM Watson API

Calling the question service DTU IBM Watson instance with Python:

```
import json, requests

user, password = 'dtu', '6H8hj3kJJ'   # These are not the real credentials
question = "Are you there, Watson?"
url = 'https://watson-wdc01.ihost.com/instance/517/deepqa/v1/question'
response = requests.post(url=url,
                         headers={'Content-type': 'application/json',
                                  'Accept': 'application/json'},
                         data=json.dumps({"question": {"questionText": question}}),
                         auth=(user, password))
```

'517' is the DTU instance. The questions are encoded in JSON and the response is also in JSON:

```
>>> for evidence in response.json()['question']['evidencelist']:
...     print(evidence['text'][:80])
Those are both pretty popular. To some extent, a plain old styling/curling brush
And if you're rejected, don't let it bother you. You can feel confident that you
First of all, you cannot make a list of the things you want in a life partner an
...
```

IBM Watson API: Question analysis

Apart from the scored documents/passagers the API also returns question analysis results, e.g., slot grammar analysis, WordNet relations, question classification, . . .

With the question “Who played drums on ’Love me do’?” posed to the question service the following is returned (here represented in YAML):

```
focuslist:
- {value: Who}
formattedAnswer: false
id: 9A95C89CD0CA4961A8BF0EDF951603E2
items: 5
latlist:
- {value: Who}
passthru: ''
pipelineid: '1230804912'
```

```
qclasslist:
- {value: DESCRIPTIVE}
- {value: FACTOID}
- {value: DEFINITION}
questionText: Who played drums on 'Love me do'?
status: Complete
synonymList:
- lemma: play
  partOfSpeech: verb
  synSet:
  - name: Lemma_Expansion
    synonym:
    - {isChosen: true, value: play, weight: 1.0}
  value: played
- lemma: drum
  partOfSpeech: noun
  synSet:
```

Watson

```
    - name: Wordnet_drum-noun-1
      synonym:
      - {isChosen: true, value: membranophone, weight: 1.0}
      - {isChosen: true, value: tympan, weight: 1.0}
    - name: Lemma_Expansion
      synonym:
      - {isChosen: true, value: drum, weight: 1.0}
    value: drums
  - lemma: do
    partOfSpeech: verb
    synSet:
    - name: Wordnet_make-verb-1
      synonym:
      - {isChosen: true, value: make, weight: 1.0}
    value: do
```

IBM Watson API question classification

IBM Watson Jeopardy!-version had the following classes: definition, category-relation, "fill-in-the-blank", abbreviation, puzzle, etymology, verb, translation, number, bond, multiple-choice, date (Lally et al., 2012).

API returns "qclasslist (string, optional): The container for a list of question classes that are determined by the pipeline for the final answer."

The API seems to return somewhat different question classes:

```
$ python -m brede.api.watson --yaml "Who played drums on 'Love me do'?" \
    | egrep -A 3 qclasslist
  qclasslist:
  - {value: DESCRIPTIVE}
  - {value: FACTOID}
  - {value: DEFINITION}
```

...IBM Watson API question classification

Another question classification example:

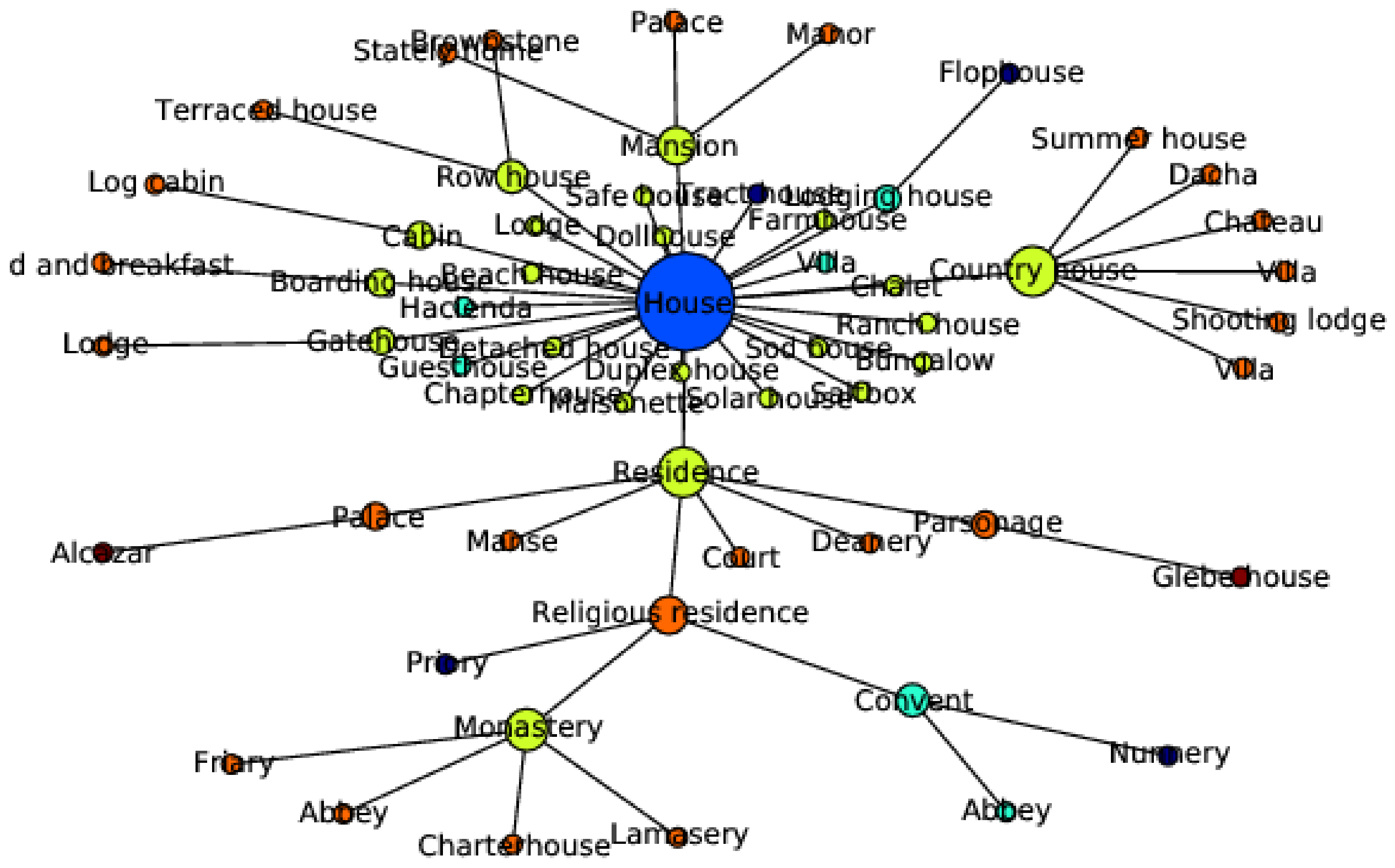
```
$ python -m brede.api.watson --yaml "What does the acronym ALS stand for?"
    | egrep -A 2 qclasslist
  qclasslist:
  - {value: FACTOID}
  - {value: DESCRIPTIVE}
```

And yet another:

```
$ python -m brede.api.watson --yaml "How do I clean the window of my car" \
    | egrep -A 5 qclasslist
  qclasslist:
  - {value: DESCRIPTIVE}
  - {value: FACTOID}
```

Definition, descriptive, factoid and focusless seem to be the classes reported from the API.

WordNet



Inspired from (Bird et al., 2009, section 4.8, pages 169+)

. . . WordNet

Using Linux command-line interface:

```
$ wn gatehouse -over

Overview of noun gatehouse

The noun gatehouse has 1 sense (no senses from tagged texts)

1. gatehouse -- (a house built at a gateway; usually the
                 gatekeeper's residence)
```

From within Python:

```
>>> from nltk.corpus import wordnet as wn
>>> for synset in wn.synsets('gatehouse'):
...     print(synset.name(), synset.definition())
(u'gatehouse.n.01', u"a house built at a gateway; usually the gatekeeper's
```

IBM Watson API and WordNet

```
$ python -m brede.api.watson --yaml "Who played drums on 'Love me do'?" \
    | egrep -A 10 "lemma: drum"
  - lemma: drum
    partOfSpeech: noun
    synSet:
    - name: Wordnet_drum-noun-1
      synonym:
      - {isChosen: true, value: membranophone, weight: 1.0}
      - {isChosen: true, value: tympan, weight: 1.0}
    - name: Lemma_Expansion
      synonym:
      - {isChosen: true, value: drum, weight: 1.0}
    value: drums
```

From within Python:

```
>>> Counter([lemma for synset in wn.synsets('drums')
                   for lemma in synset.lemma_names()]).most_common(5)
[(u'drum', 9), (u'tympan', 1), (u'membranophone', 1),
 (u'drumfish', 1), (u'metal_drum', 1)]
```

IBM Watson API and part-of-speech tagging

Part-of-speech (POS) tags returned from the API:

```
$ python -m brede.api.watson --yaml "Who played drums on 'Love me do'?" \
    | egrep -B 1 "partOfSpeech"
  - lemma: play
    partOfSpeech: verb
--
  - lemma: drum
    partOfSpeech: noun
--
  - lemma: do
    partOfSpeech: verb
```

The POS-tags seem only to be returned for the synonyms for nouns and verbs found with WordNet: `synset.pos()`.

Note that 'love' is not identified.

Python part-of-speech tagging

Finding nouns and verbs in Python with spaCy ‘somewhat equivalently’:

```
>>> from __future__ import unicode_literals
>>> from spacy.en import English
>>> text = "Who played drums on 'Love me do'?"
>>> nlp = English()
>>> tokens = nlp(text)
>>> [(token.orth_, token.lemma_, token.tag_)
         for token in tokens
         if token.tag_[:2] in ('NN', 'VB')]
[(u'played', u'play', u'VBD'), (u'drums', u'drum', u'NNS'),
 (u'Love', u'love', u'NN'), (u'do', u'do', u'VB')]
```

Focus and LAT

Lexical answer types (LAT):

```
$ python -m brede.api.watson --yaml "In what city or country did
          John Lennon meet Yoko Ono?" | egrep -A 2 latlist
  latlist:
  - {value: city}
  - {value: country}
```

Focus ("the part of the question that is a reference to the answer", (Lally et al., 2012))

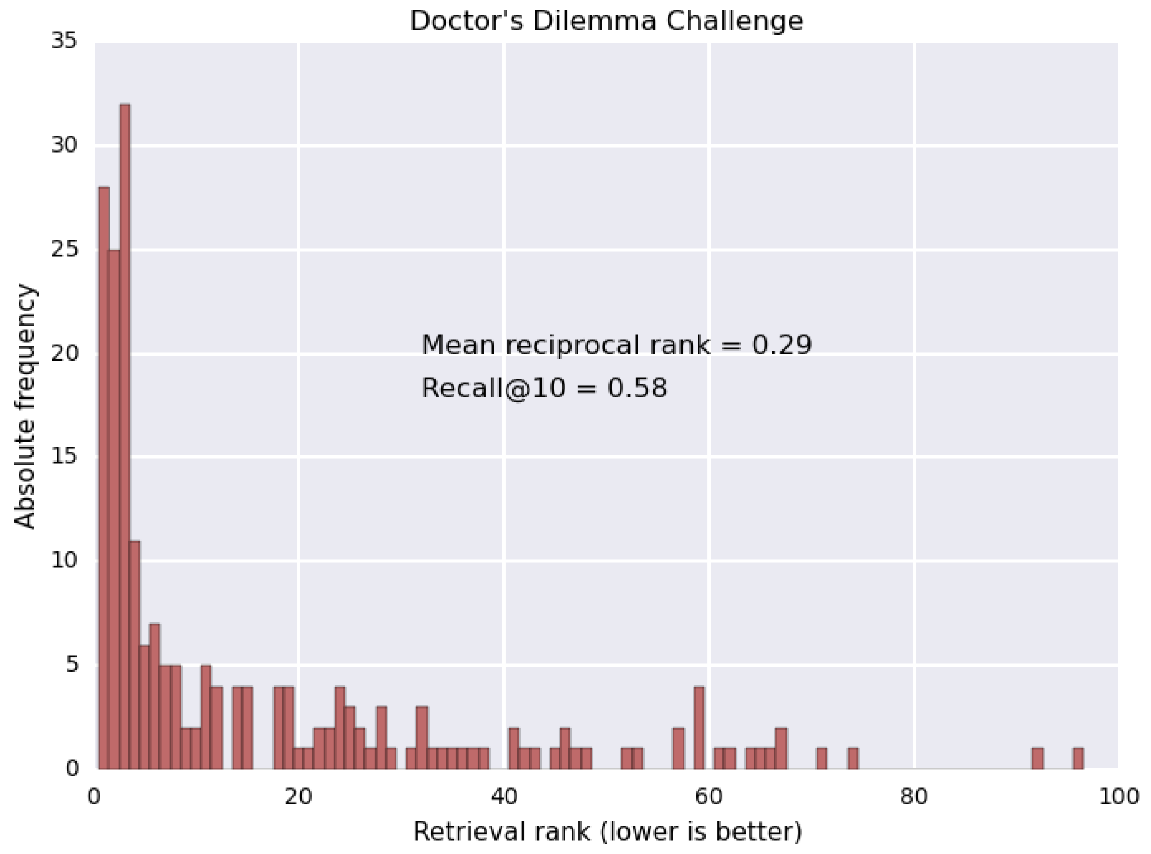
```
$ python -m brede.api.watson --yaml "In what city or country did
          John Lennon meet Yoko Ono?" | egrep -A 1 focus
  focuslist:
  - {value: what city or country}
```

Focus and LAT

Results returned from the API based on a query sentence:

Sentence	Focus	LAT	QClass
How do I wash my car?	how	—	descriptive, factoid
Ability to probe to the bone in a diabetic foot ulcer predicts this disease	—	disease	descriptive, focusless
Did John Lennon meet Yoko Ono in London?	—	—	focusless, descriptive
Where did John Lennon meet Yoko Ono?	where	where	factoid, descriptive
Which drummer recorded 'Love me do'?	which drummer recorded	drummer	descriptive, factoid

Some results



IBM's test of Doctor's Dilemma questions with Watson: Recall@10: 0.77 (Ferrucci et al., 2013)

FindZebra' mean reciprocal rank on 56 questions: 0.385 (Drăgusin et al., 2013)

IBM Watson (211 unseen questions): Mean reciprocal rank: 0.2899, Recall@10: 0.5829.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly, Sebastopol, California. ISBN 9780596516499. *The canonical book for the NLTK package for natural language processing in the Python programming language. Corpora, part-of-speech tagging and machine learning classification are among the topics covered.*

Chu-Carroll, J., Fan, J., Schlaefer, N., and Zadrozny, W. (2012). Textual resource acquisition and engineering. *IBM Journal of Research and Development*, 56(3/4):4. DOI: 10.1147/JRD.2012.2185901. *Description of the unstructured corpora collected and used in the IBM Watson Jeopardy! system.*

Drăgusin, R., Petcu, P., Lioma, C., Larsen, B., Jørgensen, H. L., Cox, I. J., Hansen, L. K., Ingwersen, P., and Winther, O. (2013). Findzebra: a search engine for rare diseases. *International Journal of Medical Informatics*, 82(6):528–538. DOI: 10.1016/j.ijmedinf.2013.01.005.

Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., and Mueller, E. T. (2013). Watson: Beyond Jeopardy. *Artificial Intelligence*, 199-200:93–105. DOI: 10.1016/j.artint.2012.06.009.

Lally, A., Prager, J. M., McCord, M. C., Boguraev, B. K., Patwardhan, S., Fan, J., Fodor, P., and Chu-Carroll, J. (2012). Question analysis: how Watson reads a clue. *IBM Journal of Research and Development*, 56(3.4). DOI: 10.1147/JRD.2012.2184637.