Swiss Re

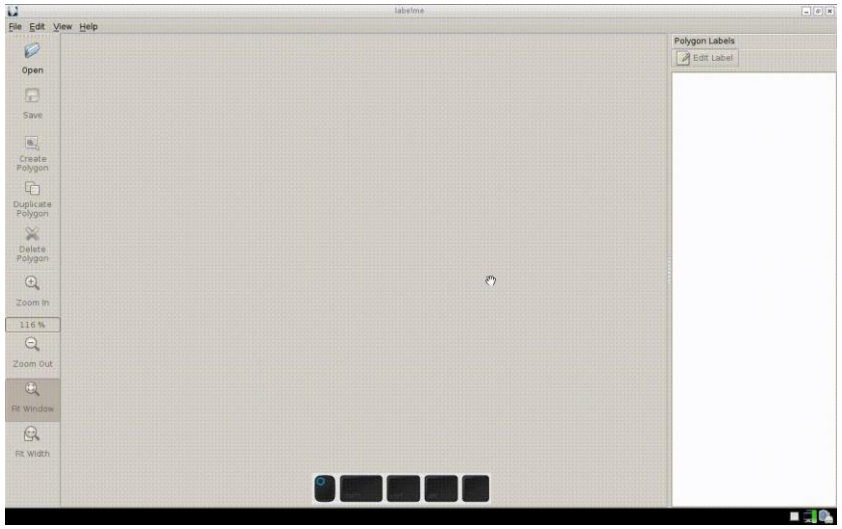
Annotation Tools and Beyond

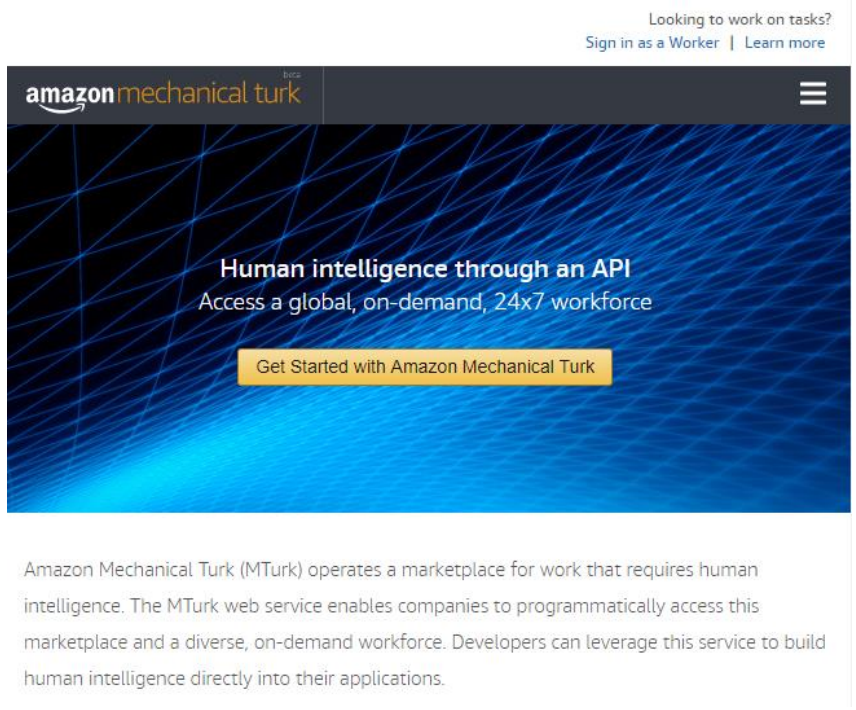
Guan Wang
2018/06/06

Table of Contents / Agenda

Why Annotation Tools?

- Most ML Applications are **Supervised** Learning
- If lucky: automatically generated
  - Click Google ad on a webpage
  - Add stuff in shopping cart on Amazon
  - Listen to music on Spotify
- Not so lucky, but there are smart stricks
  - Crawl from Internet
  - Distant Supervision
- Not lucky, no tricks (usually)
  - Label by hands



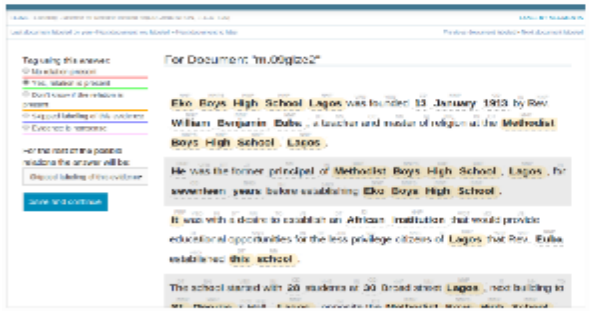


Annotator for Chinese

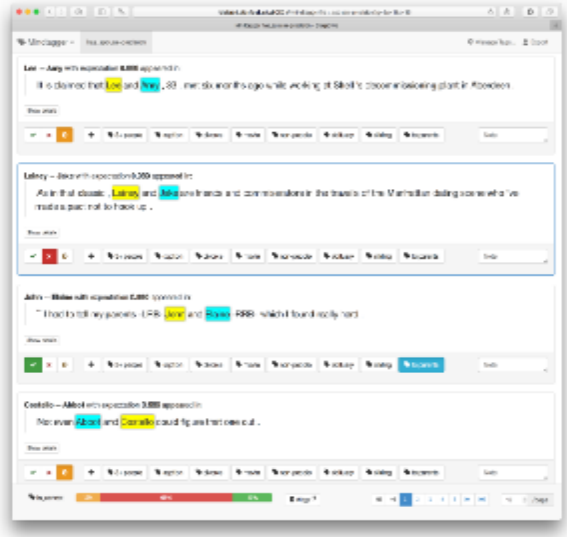
- Many tasks in NLP are supervised learning problem
  - Sequence Labelling (Tokenization, NER)
  - Classification (relation extraction, sentiment analysis, intent recognition)
- Not so much open-source annotated corpus for Chinese
- Vertical Applications require **Domain Knowledge** (insurance, finance, health, legal, public security etc.)
- Existing annotation tools are either
  - clumsy and complex to use
  - only supports English
  - not open-source and cannot use without public cloud
  - use out-dated software stacks

Annotator for Chinese

IEPY



DeepDive (Mindtagger)

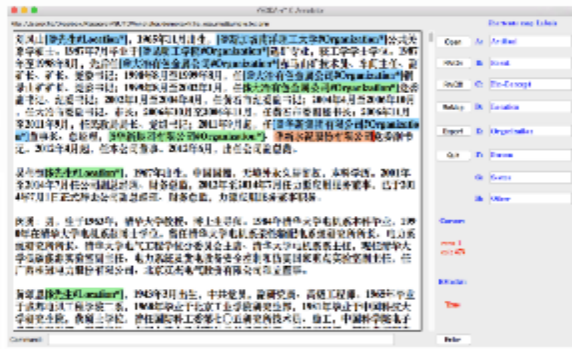


Snorkel

Figure 3: Labeling functions which express pattern-matching, distant supervision, and weak classifier heuristics, respectively, in Snorkel's Jupyter notebook interface.

BRAT

SUTDAnnotator



Slate

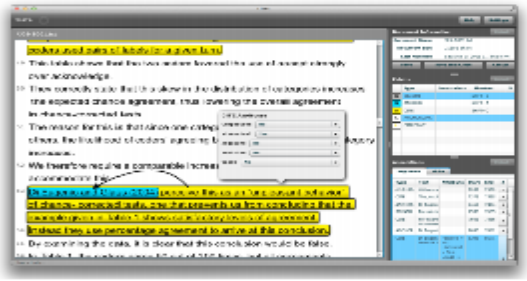
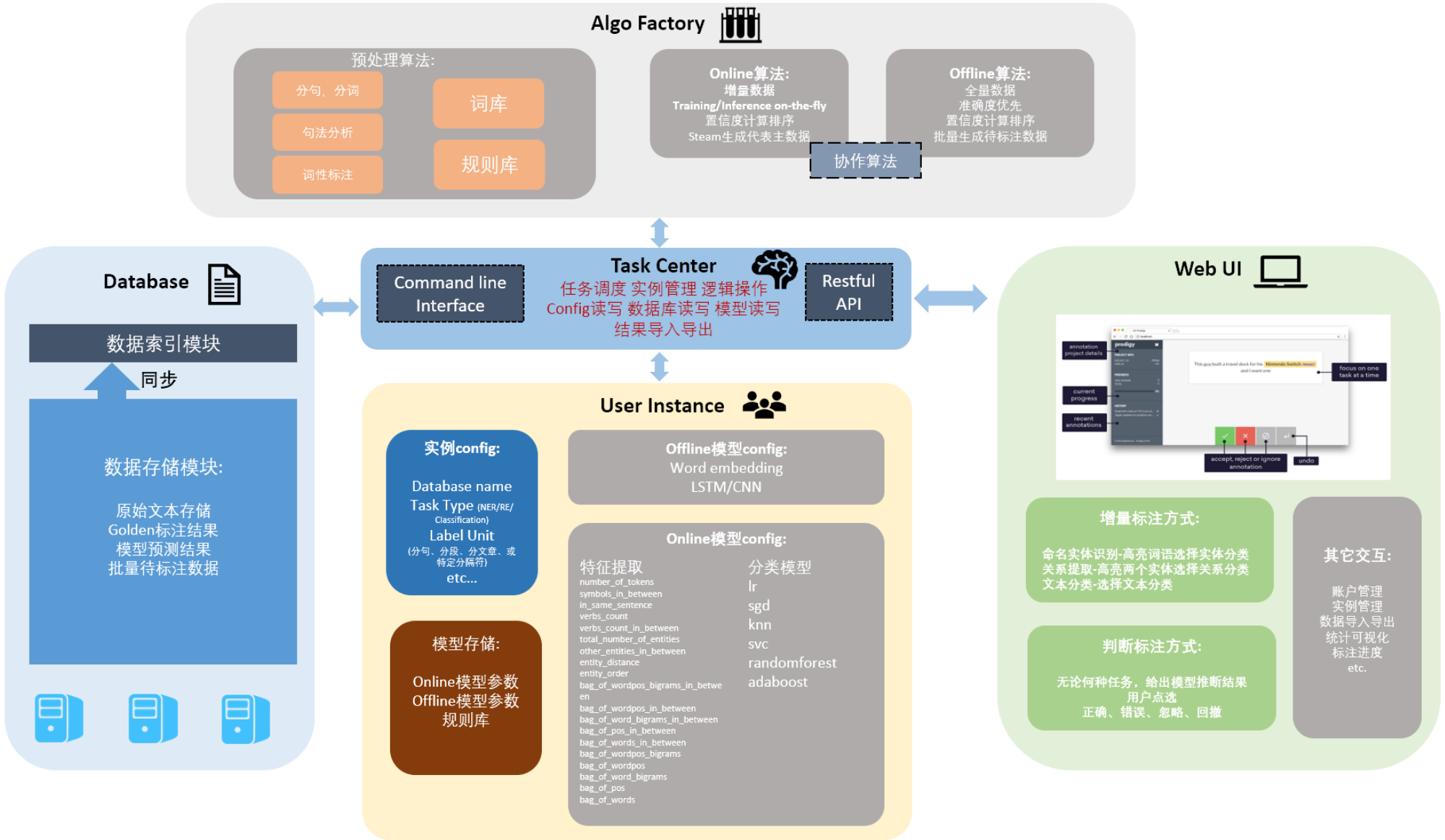


Figure 4: The Viewer utility in Snorkel, showing candidate company-employee relation mentions, comprised of candidate person and company mention pairs.

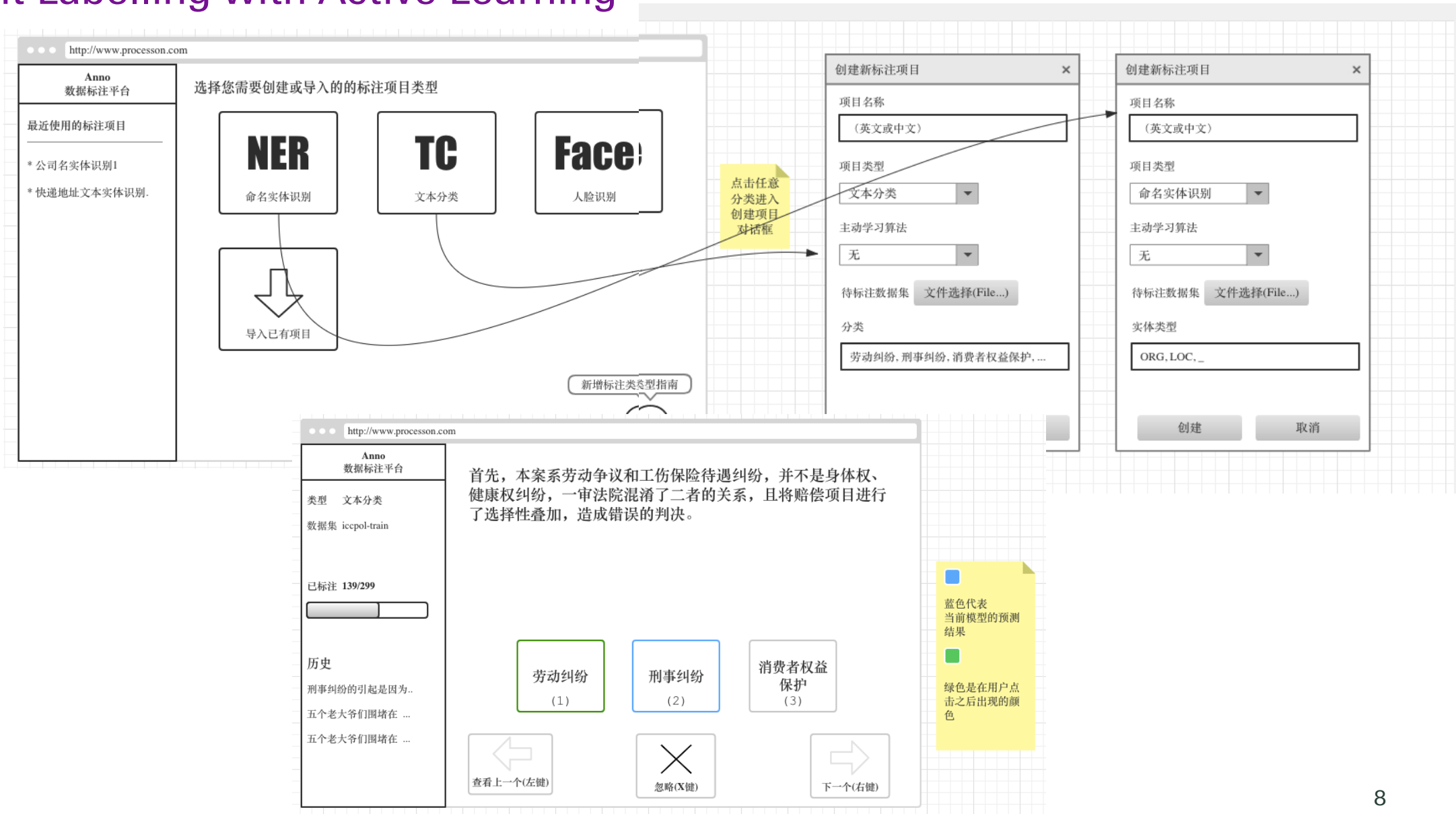
Intelligent Labelling with Active Learning

1. User make labels
2. Backend active learning algorithm will consist of "Online" part and "Offline" part.
    - "Online" part will do the online learning and update online model in real time, using fast traditional algorithms like Linear Classifiers or SVM
    - When label data accumulated to a certain amount, "Offline" part will update the offline model, using probably highly accurate deep learning models.
3. After model is updated, predict as much as possible within reasonable time, rank the confidence, and choose the lowest certain number of samples as datasets waiting to be labeled. Repeat step 1.

Intelligent Labelling with Active Learning

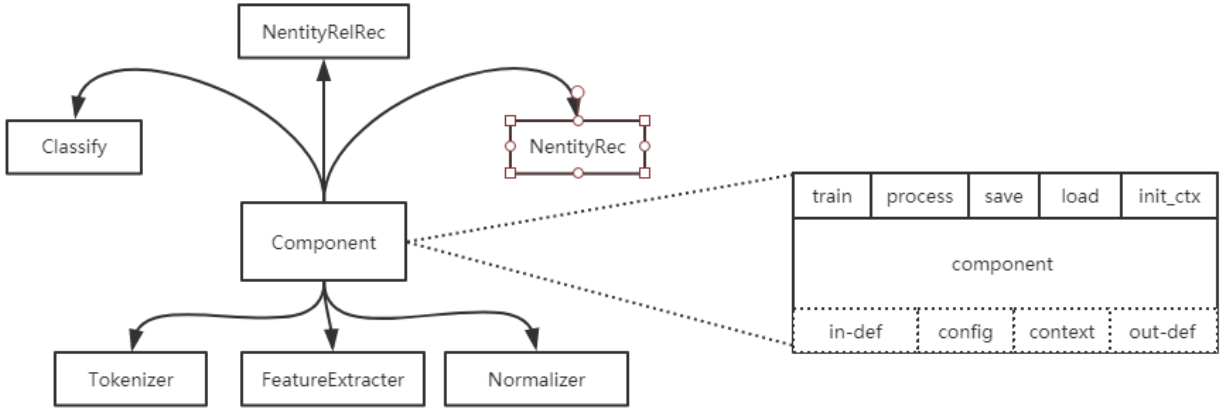


Intelligent Labelling with Active Learning

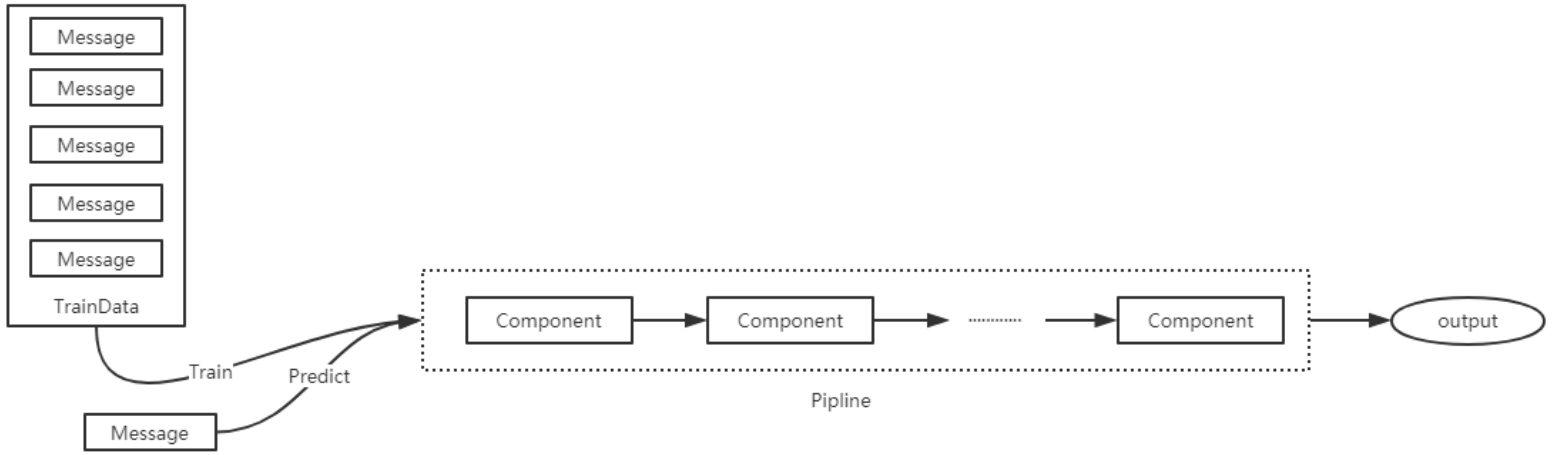


Data Pipeline and Modular Design

Modular Design



Data Pipeline Design



Data Pipeline and Modular Design

Configurable Programming

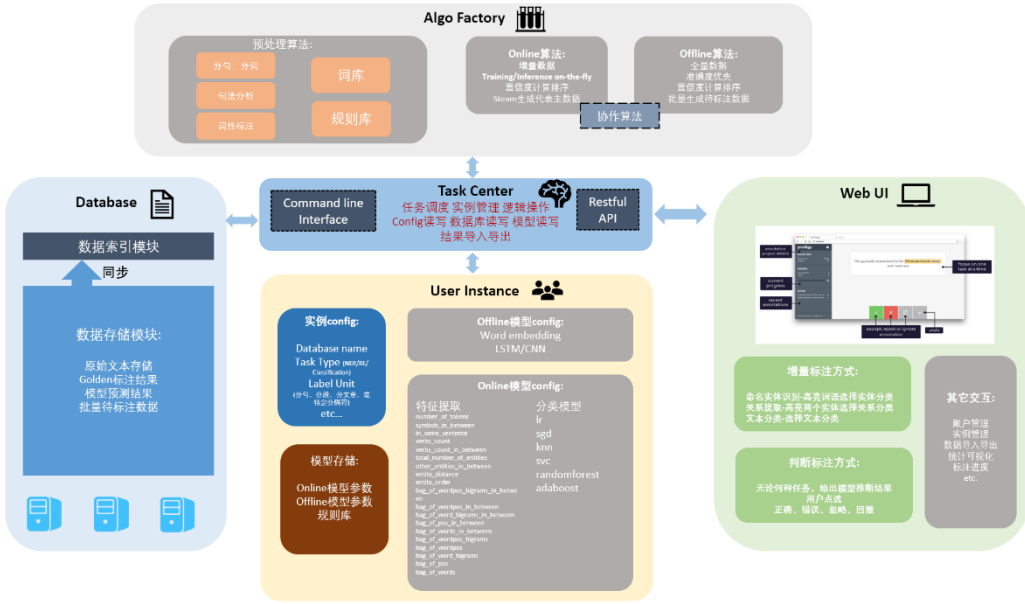
```
{
 "ip" : "localhost",
 "port": "8000",
 "database_type": "mongodb",
 "type": "classification"
 "name": "email_spam_classification",
 "model_type" : "classification",
 "pipeline": ["nlp_word2vec",
              "linesplit_preprocess",
              "feature_extractor",
              "online_svm_classifier_sklearn",
              "offline_svm_classifier_sklearn"],
 "language": "zh",
 "wordvec_file": "./tests/data/test_embedding/vec.txt",
 "path" : "./tests/models",
 "org_data" : "./tests/data/test_email_classify/email_classify_chi.txt",
 "database_name" : "spam_emails_chi",
 "labels": ["spam","notspam"],
 "batch_num" : "10",
 "inference_num" : "20",
 "low_conf_num" : "10",
 "confidence_threshold" : "0.95",
 "log_level": "INFO",
 "log_file": null
}
```

More than an Annotation Tool

Thanks to **Modular** and **API** design:

1. Human User Interface for Machine Learning Projects
2. Data Manager
   - raw data, pre-processed data, feature engineered data, labelled data, predicted data etc.
   - Upstream module like Crawlers
   - Downstream modules like Visualizations
3. Model Manager
   - pre-trained models, configuration of online and offline models, persisted models
4. Prediction Service

Full Pipeline Machine Learning Tool



One More Thing

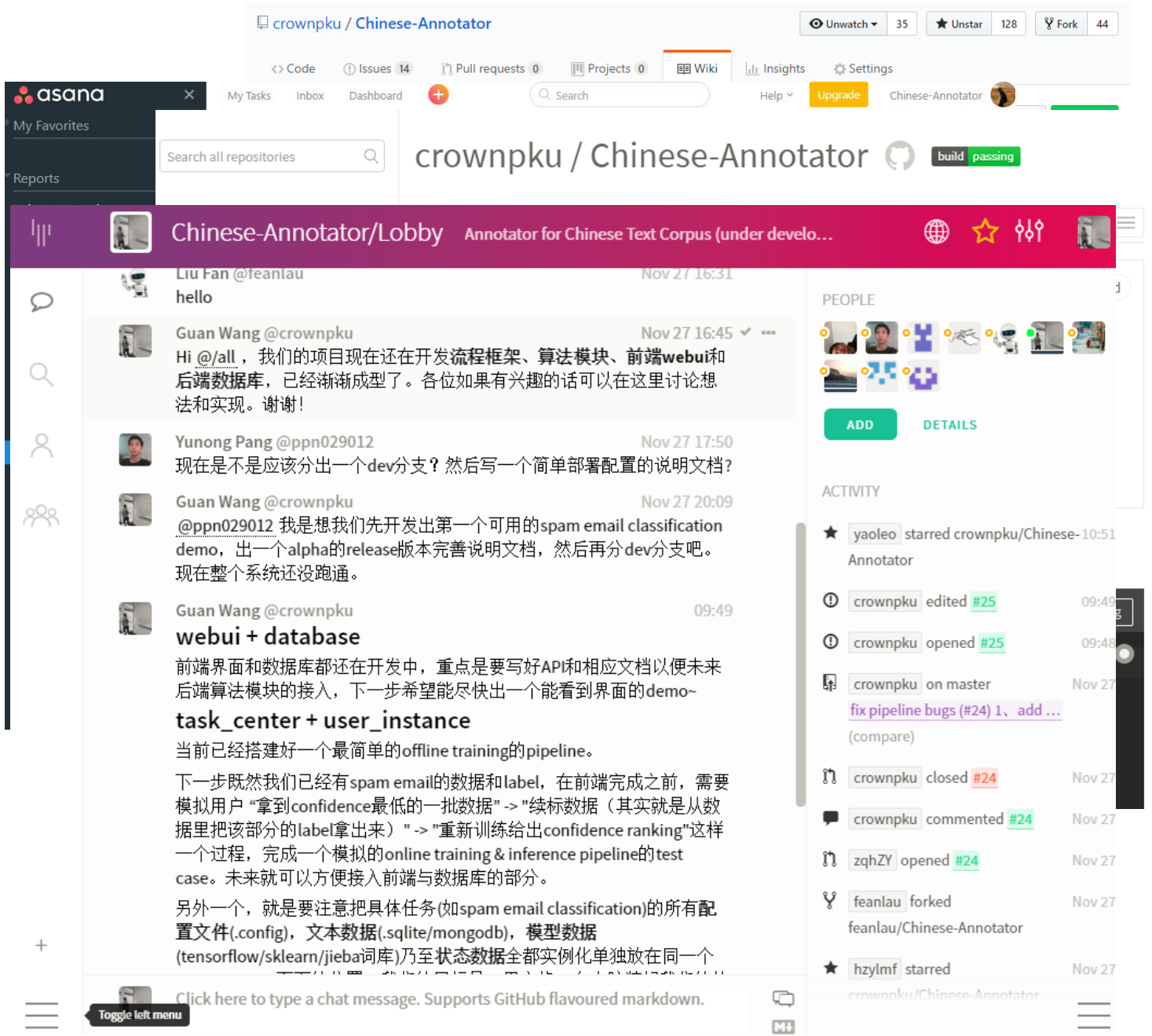
- 10 People
- Full Stack hackers, algorithm experts, professional software engineers
- Scattered in Chengdu, Shanghai, Nanjing, Beijing, Guangzhou, Shenzhen and Hong Kong (luckily same time zone)
- Process
  - Vague idea
  - White Board design
  - Software architecture diagram
  - Choice of stack back/front end
  - Configurable and Pluggable Algorithm Design
  - Unit Testing Cases

- As of Feb.22nd 2018:
  - 30k lines of codes
  - 153 commits
  - 248 stars
  - 76 forks
  - 14 issues
  - 41 pull requests.

One More Thing

Tools we use:

- Github
  - Code review
  - issue tracker
  - code discussion
  - wiki documents
  - Travis-CI (continuous integration)
- Asana
  - project management
  - allocate and claim tasks
  - set deadlines
  - synchronize progress
  - share resoures
  - discuss spedific tasks
- Wechat (core team discussion)
- Gitter chatroom (public discussion)



One More Thing

- Working Style: **Distributed Asynchronous Collaboration**
- No compulsory meetings, every communication is online and persisted in an asynchronous way
  - Pull requests at mid-night
  - Merge and review code on the subway
- Every member chooses when, where and how he works.
  - beds, sofas, cafes, libraries, swimming pool or anywhere with good quality Wifi. No commute.
  - Save rent; Help environment protection
- Flat and Transparent, Spontaneous Collaboration, Full Trust and Support for Human Nature

Q&A

Legal notice

The information and opinions contained in the presentation are provided as at the date of the presentation and are subject to change without notice. Although the information used was taken from reliable sources, Swiss Re does not accept any responsibility for the accuracy or comprehensiveness of the details given. All liability for the accuracy and completeness thereof or for any damage or loss resulting from the use of the information contained in this presentation is expressly excluded. Under no circumstances shall Swiss Re or its Group companies be liable for any financial or consequential loss relating to this presentation.