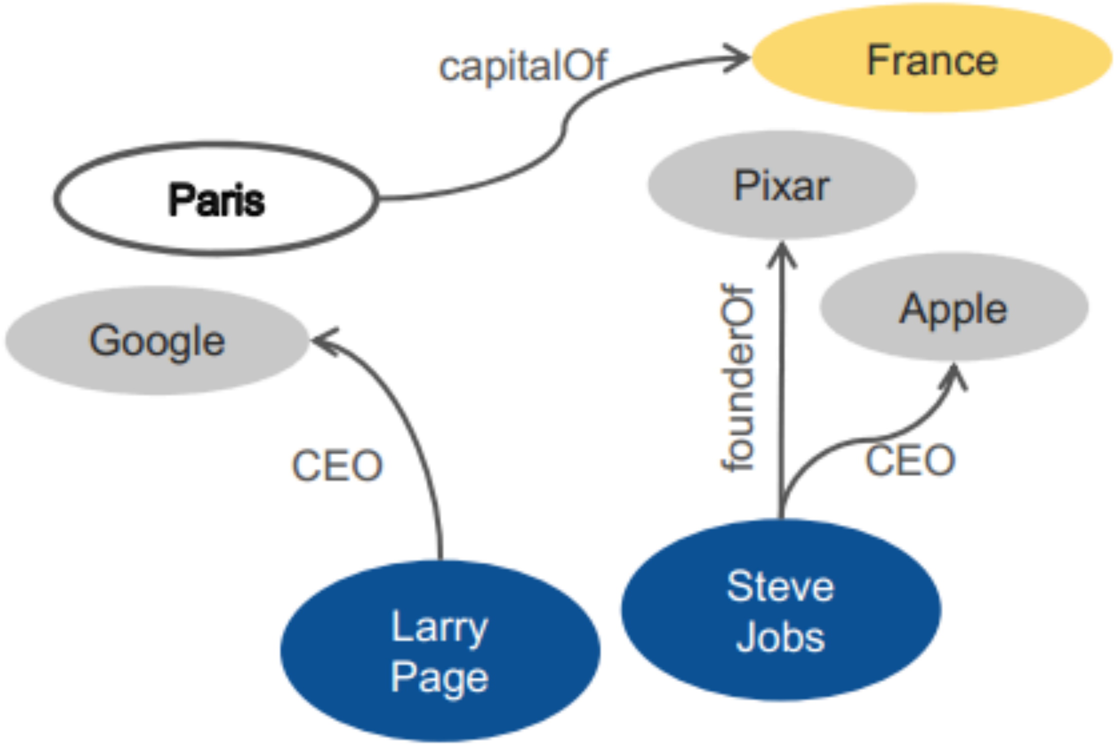
Exploring the Efficiency of Batch Active Learning for Human-in-the-Loop Relation Extraction

Ismini Lourentzou
UIUC
lourent2@illinois.edu

Daniel Gruhl
IBM Research Almaden
dgruhl@us.ibm.com

Steve Welch
IBM Research Almaden
welchs@us.ibm.com

Extract relations of interest from free text



http://www.mathcs.emory.edu/~dsavenk/slides/relation_extraction/img/distant.png

Google CEO Larry Page announced that...

Steve Jobs has been Apple for a while...

Pixar lost its co-founder Steve Jobs...

I went to Paris, France for the summer...

Useful for:

- knowledge base completion
- social media analysis
- question answering
- …

Extract relations of interest from free text

Task: binary (or multi-class) classification

sentence $S = w_1\ w_2 .. \mathbf{e_1} .. w_j .. \mathbf{e_2} .. w_n$

$\mathbf{e_1}$ and $\mathbf{e_2}$ entities

The new **iPhone 7 Plus** includes an improved **camera** to take amazing pictures

Component-Whole(e_1 , e_2) ?

YES / NO

It is also possible to include more than two entities as well:
"At codons 12, the occurrence of point mutations from G to T were observed" → point mutation(**codon**, **12**, **G**, **T**)

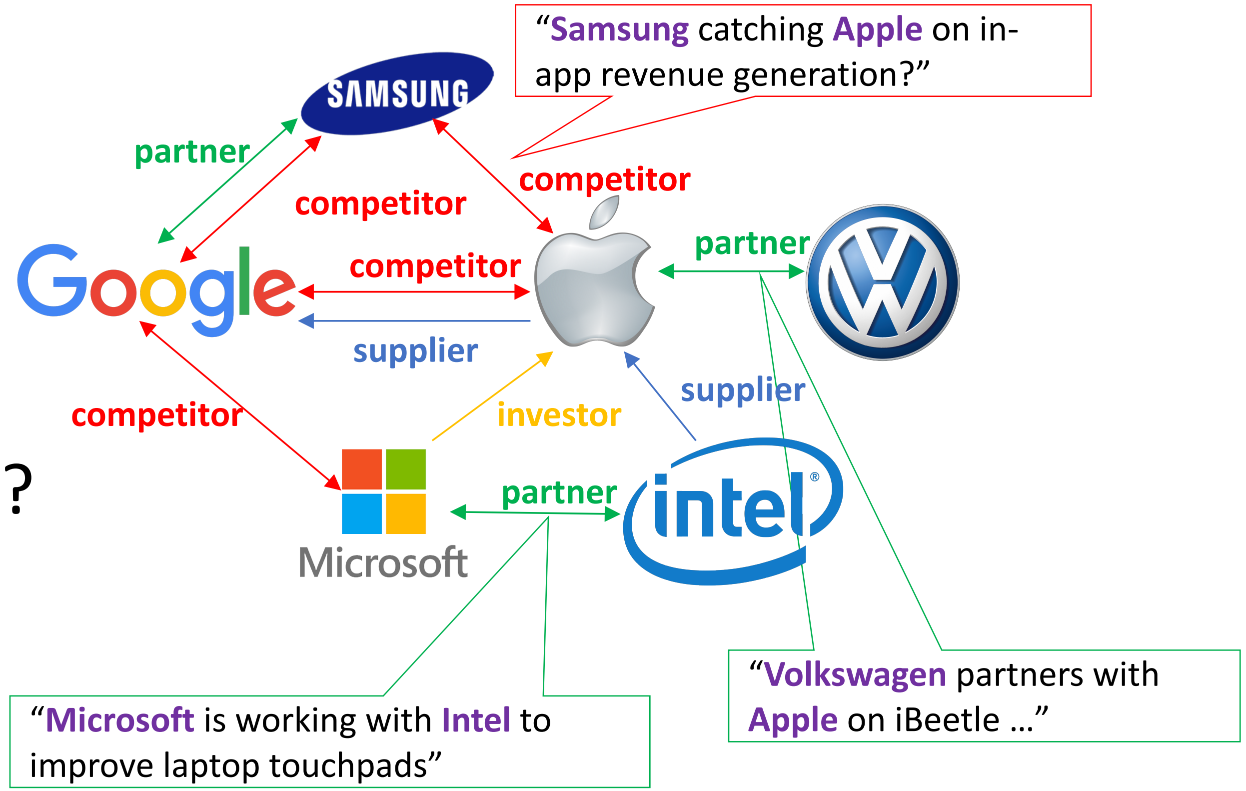
Challenge: "On-demand" Relation Extraction

Most NLP applications require domain-specific knowledge

Assist in strategic company marketing

Which companies supply Google?

Who is the biggest competitor of Apple?



Challenge: "On-demand" Relation Extraction

Most NLP applications require domain-specific knowledge

Ideally, we aim to achieve:

- ✓ fast training of any relation
- ✓ according to user-defined requirements
- ✓ under limited annotated data
- ✓ not relying on additional knowledge sources
    - linguistic structured or textual

Recent state of the art on relation extraction has been focusing on …

- Incorporating linguistic knowledge in (neural) architectures
- Maximizing performance by means of feature engineering

Requisite: availability of large datasets

Unfeasible!

expensive & challenging to acquire large amounts of reliable gold standard training data

the definition of a relation is highly dependent on the task at hand and on the view of the user

https://edumine.files.wordpress.com/2015/04/searching-insanely-large-datasets.jpg

Distant supervision

Exploit large knowledge bases to automatically label entities in text

Assumption: when two entities co-occur in a sentence, a certain relation is expressed

KB

Relation	Entity 1	Entity 2
place of birth	**Michael Jackson**	**Gary**
place of birth	**Barack Obama**	**Hawaii**
…	…	…

text

Barack Obama moved from **Gary** ….

Michael Jackson met … in **Hawaii**

place of birth

False positives and low tail coverage!

For many ambiguous relations, co-occurrence does not guarantee the existence of the relation

Multi-instance learning methods cannot handle sentence-level prediction or bags where all sentences do not describe a relation.

Frequent entities/relations will have good coverage, tail ones may not be well represented.

Active Learning

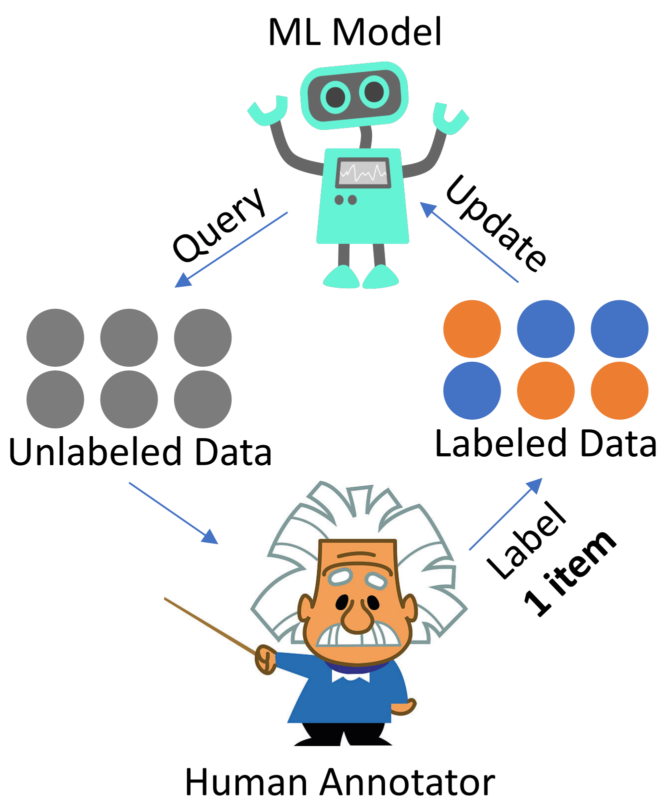
Find the most **efficient** way to query unlabeled data and learn a classifier with the minimal amount of human supervision.

Sequential active learning: single instance at each iteration

When training takes a long time (e.g., **NNs**)

- updating the model after each label is costly
  - human annotation time: waiting for the next datum to tag
  - time to update the model and select the next example
  - computing resources

- When local optimization methods are used (e.g., **NNs**)
  - highly unlikely a single point to result in significant impact on the performance



Batch Active Learning

Batch active learning: select a batch of instances at each iteration

Trade-off between efficiency and performance

- Large batches result in…
  - Less frequent model updates
  - Increased prediction error

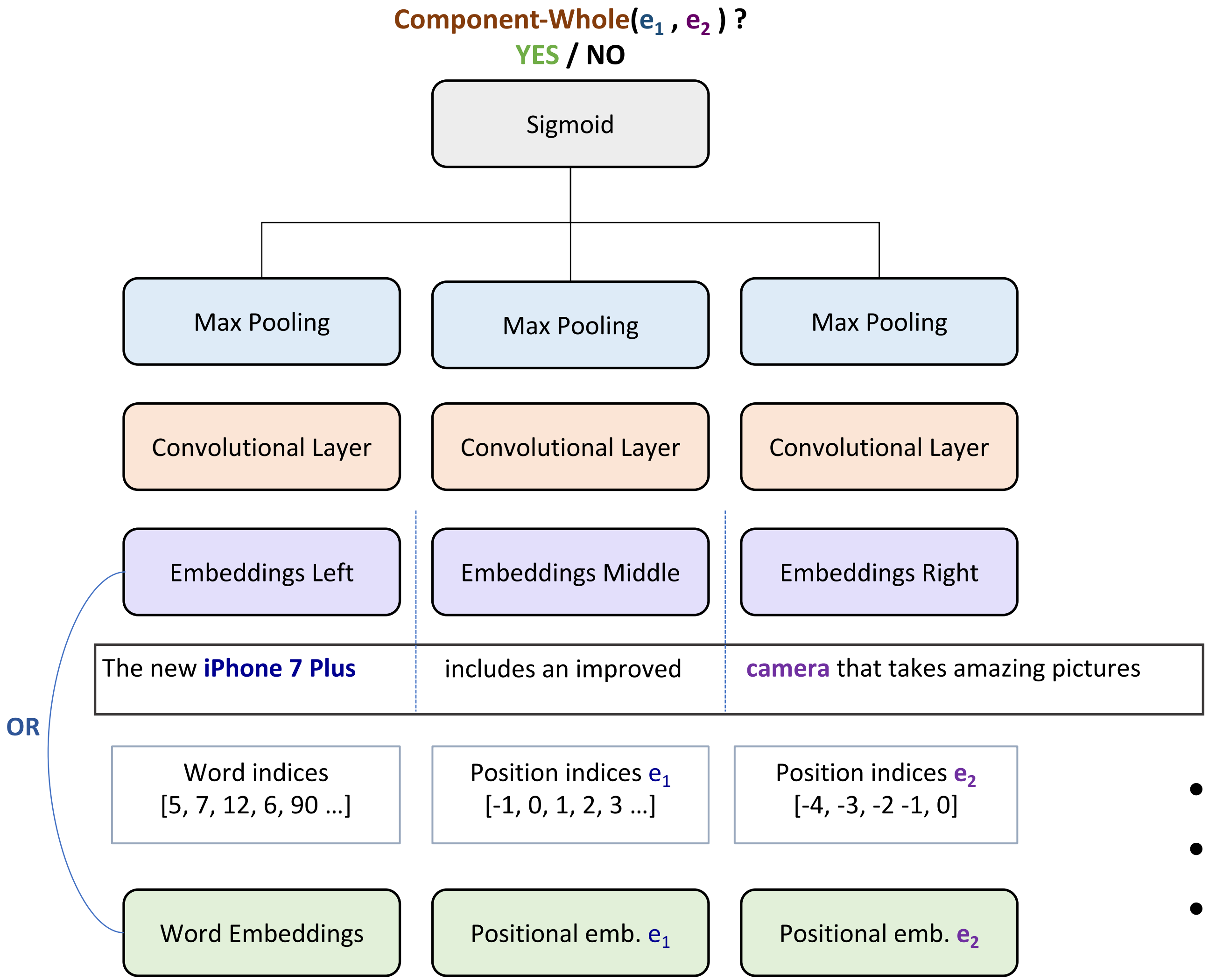
http://fredgolfrange.com

Let's explore this trade-off!

- Train neural models
- For extracting arbitrary user-defined relations
- From potentially infinite pool of unlabeled Web and social stream data

Ultimate goal: optimize batch size **+** satisfactory performance **+** reduce total training time

Our models and AL methods



Convolutional Neural Networks (CNNs) because:

- ✓ highly expressive leading to low training error
- ✓ faster in training than recurrent architectures
- ✓ known to perform well in relation classification

1. **CNNpos**: word sequences and positional features
2. **CNNcontext**: context-wise split sentence

Active Learning methods

- **us**: (uncertainty) ranking based on model confidence
- **quire**: informativeness + representativeness
- **bald**: Monte Carlo + Dropout for uncertainty

Evaluation datasets

Dataset	#examples	Relations
Semeval10 Task 8	10,717	9 types: Entity-Origin, Message-Topic, etc.
CausalADEs	1,420	causal drug-ADE relations from medical forum posts

Semeval10 Task 8

Cause-Effect, Component-Whole, Content-Container, Entity-Destination, Entity-Origin, Instrument-Agency, Member-Collection, Message-Topic, Product-Producer, "Other"

CausalADEs

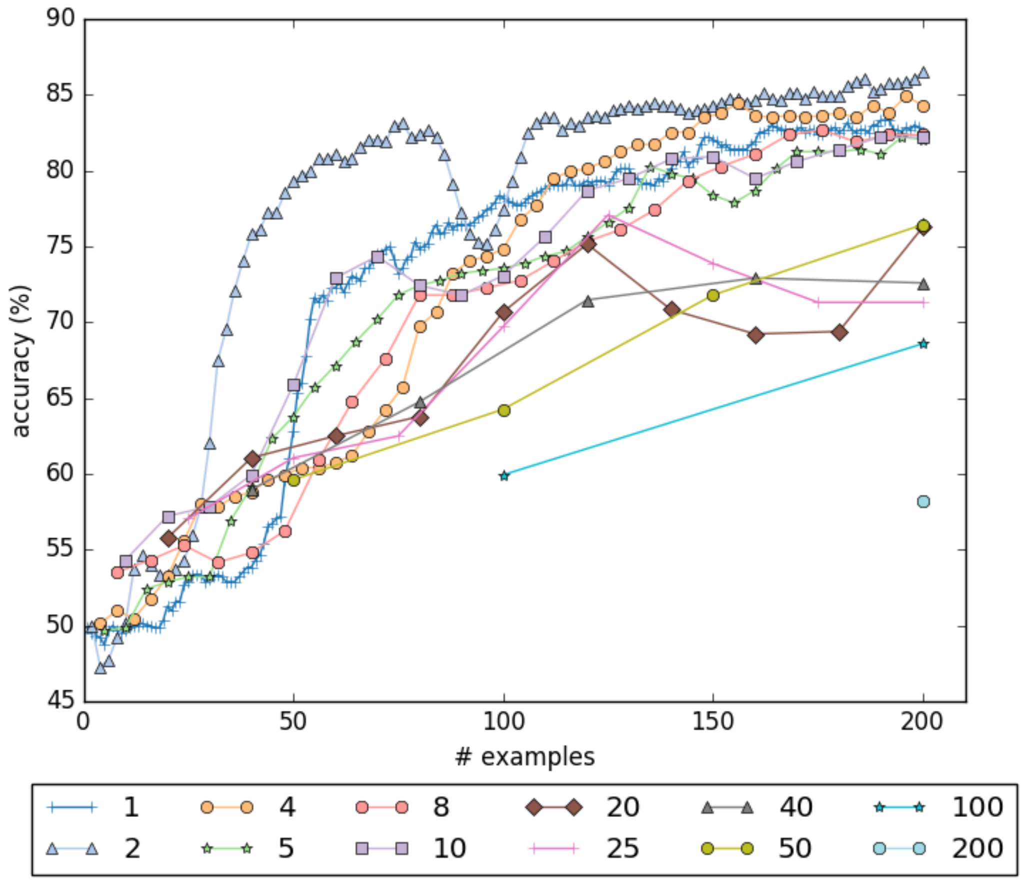
CSIRO Adverse Drug Event Corpus (Cadec)
medical forum posts on patient reported Adverse Drug Events
posts tagged based on mentions of certain drugs, ADRs, symptoms, findings etc.
We annotate a corpus similar to CADEC for **causal relationships** between drugs and ADEs

Varying the batch size in cold-start scenarios

No annotated data available

Start human annotation as quickly as possible

- Bigger batch → lower performance
- Small increase on the batch size is okay
  - By the time you've scored 200 examples, batches of 5 or 10 do nearly as well as anything else.
- High variance on the beginning
  - We need enough examples to "span the space" and to avoid overfitting



A look at the impact of batch size on training rate for one active learning strategy, one neural structure on one task. Note that the best strategy in this case is two at a time.

… But how to select the initial batch?

Rank data based on **unsupervised** text based criteria
Select top ranked ones as initial training examples

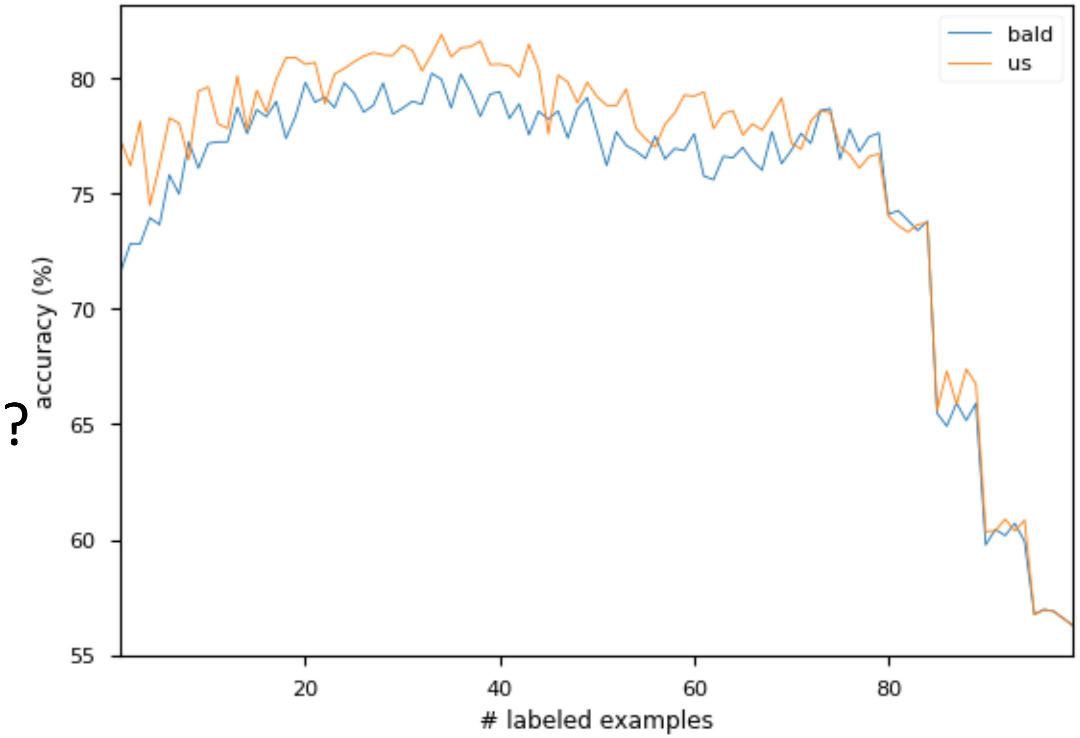
Maximize **linguistic dissimilarity (LD)** between sentences
(by utilizing Glove embeddings)

How large initial batch should be for good results?

1. Vary the size of the initial batch generated via LD
2. Fixed batch size for subsequent iterations at 5
3. Continue the process until we hit our budget constraint

Optimal initial batch ~ 30 labeled examples

< 20: overfitting initial training batch
> 40: AL unable to focus on the regions of confusion



An exploration of the impact of initial batch size.
For our datasets an an initial batch of 30 seems like a good place to start. This plot is the average of 10 datasets with CNNcontext as our classification model

And what about batch size for next runs?

Strong preference for larger batches

Computing the next "batch" & loading it into the UI for the SME to score takes time

Larger batch: negative impact on performance

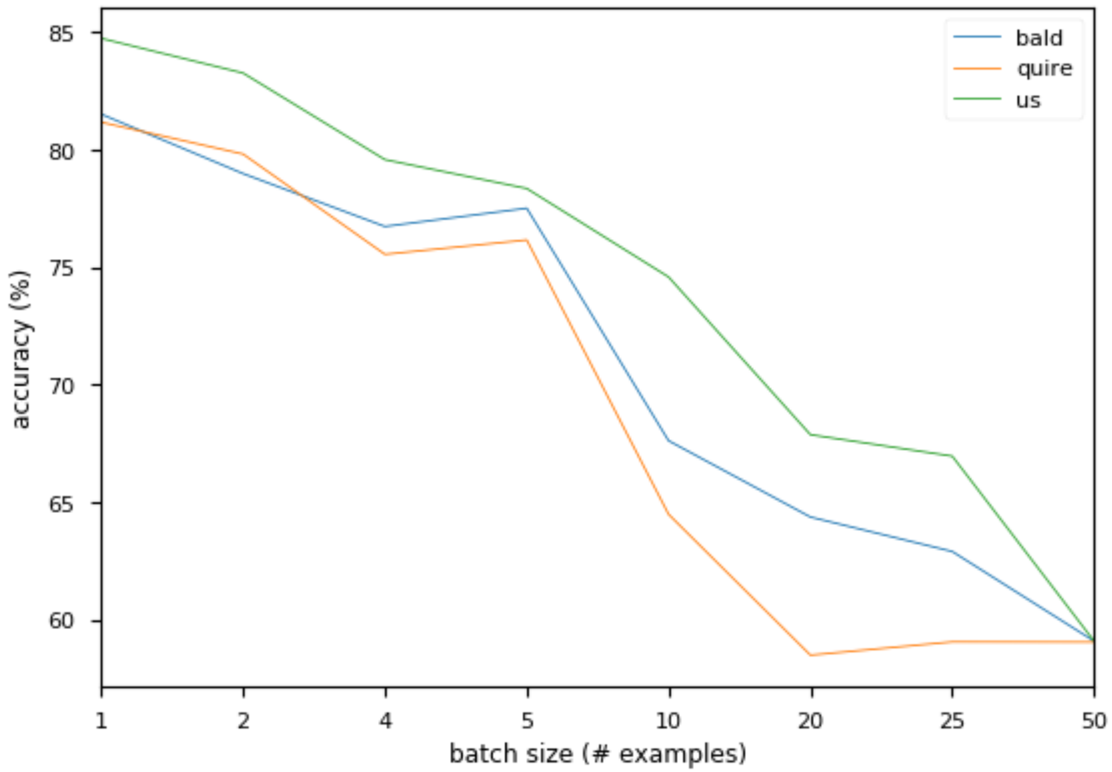
Best performance is when using batch size of 1

Real drop seems to be after 5

(which only loses 5% compared to the batch size of 1)

If your system has a finite cost associated with generating batches this may be good place to stop

A default batch size of 5 examples seems to be a good compromise between efficiency of example generation and speed of learning



CNNcontext model trained under different active learning methods. This is a look at the performance after 50 examples have been scored. Compared to the fully sequential approach of one example at a time, there is approximately only 5% decrease in the performance of using a slightly larger batch size of 5 examples.

Interleaving to reduce waiting time

Computing the next “batch” & loading it into the UI for the SME to score takes time

Workflow for a single item batch:

(1) User spends 5 seconds scoring a single example
(2) System spends 25 seconds getting next examples
(3) Repeat

Over 80% of the time the user is **waiting**!

Even with batch size of 5, 1/2 of the user time spend waiting

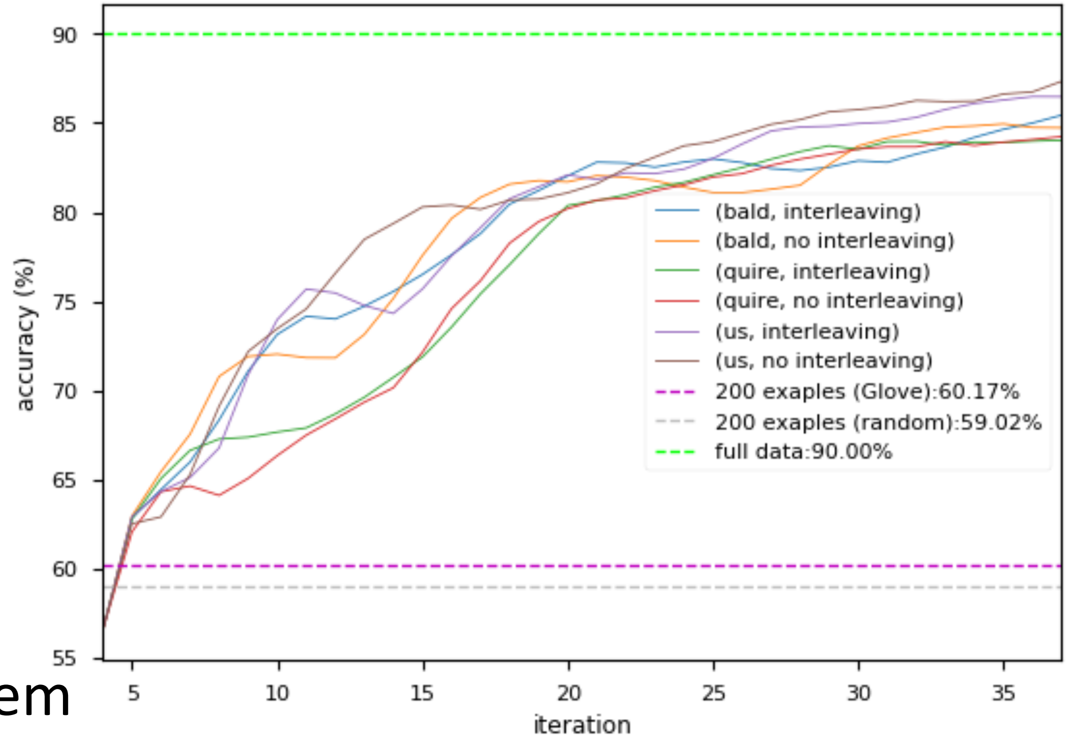
Annotation time is the **largest cost** in a HuML system

In an **ideal** world they would be **scoring constantly**.

Interleaving: Keep last unlabeled batch for future scoring

Use $B_0 \ldots B_{n-2}$ batches to produce next batch B_n

User scores batch B_{n-1} while system ranks the next batch B_n

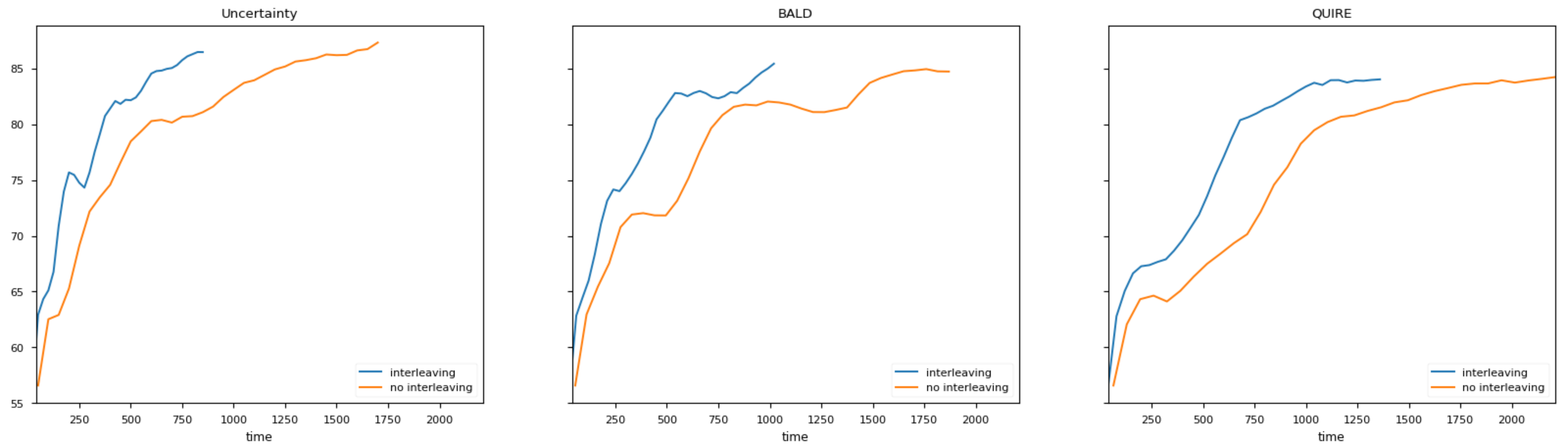


Comparison of interleaving and classic training sessions

Trained on only 20% of the data: 86% accuracy

Training with all data: 90% accuracy

Interleaving to reduce waiting time (2)



- ✓ **Continuous** human work
- ✓ Comparable performance, in **≈ 50% less training time**, irrespective of the AL method

Conclusions & Future Work

Ultimate goal: optimize batch size **+** satisfactory performance **+** reduce total training time

- Analysis of batch AL vs. sequential AL
- Competitive performance for extracting relations with very little annotated data
  - Larger initial batch size, chosen with **unsupervised curriculum learning**
  - **Interleaving** to reduce human annotation waiting time

Future work

+ Expand analysis to other tasks (we have focused on RE so far)
+ Adaptive batch size AL: dynamically update batch size between iterations
+ Non-perfect labelers: how the optimal batch size varies w.r.t. labeling noise?
+ Blending semi-supervised with batch AL
+ Meta-learning approaches, i.e. learning the best AL strategy

References

[1] I. Lourentzou, D. Gruhl, S. Welch, “Exploring the Eciency of Batch Active Learning for Human-in-the-Loop Relation Extraction”, HuML@WebConf 2018 *(this work)*

[2] I. Lourentzou, A. Gentile, D. Gruhl, A. Coden, A. Alba, S. Welch, “Mining Relations from Unstructured Content”, PAKDD 2018

[2] S. Burr, “Active learning” *Synthesis Lectures on Artificial Intelligence and Machine Learning* 2012

[3] H. Adel, B. Roth, and H. Schütze, “Comparing convolutional neural networks to traditional models for slot filling” NAACL-HLT, 2016.

For compliments e-mail
welchs@us.ibm.com

For complaints e-mail
lourent2@illinois.edu