Crowdsourcing a Statistical Language Generator using Phrase-based Factored Language Models

Francois Mairesse

Dialogue Systems Group
Machine Intelligence Laboratory
University of Cambridge

LORIA, November 23th 2010

CLASSiC

UNIVERSITY OF CAMBRIDGE

Natural Language Generation Paradigms

Templates

E.g. '*There is a flight from FROM_CITY to TO_CITY leaving at TIME on DATE*'

Simple, but does not scale

Modular architecture

Semantics → Rhetorical structure → Syntax → Realisation (Reiter & Dale, 2000)

e.g. Aggregate two syntactic trees using a relative clause

More scalable, but relies and a large set of complex rules

Difficult to maintain, as a single rule can affect a large range of outputs

→ Data-driven Language Generation

Data-driven Language Generation

Overgeneration and reranking

Reranking using SLMs (Langkilde & Knight, 99)

Ranking models trained on user feedback (Walker et al. 02; Stent et al. 04)

Learn the parameters of a handcrafted generator

Learn a PCFG from Treebank data (Belz, 08)

Reinforcement learning (Rieser & Lemon, 10)

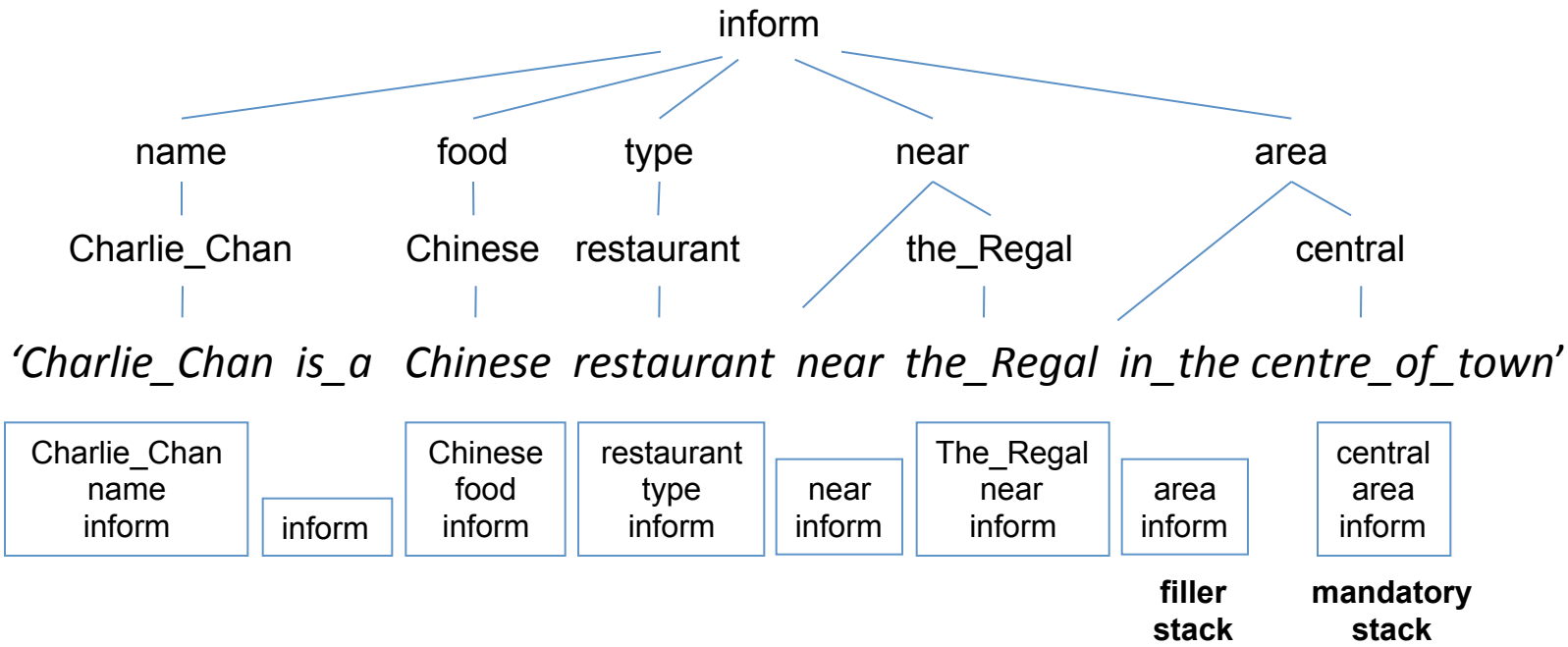
Can we learn a generator from labelled data *without* handcrafting any rules?

Semantic Representation

Data: corpus of dialogue utterances with stack-based semantics

e.g. System dialogue act

inform(name(Charlie_Chan) type(restaurant) food(Chinese) area(central) near(the_Regal))



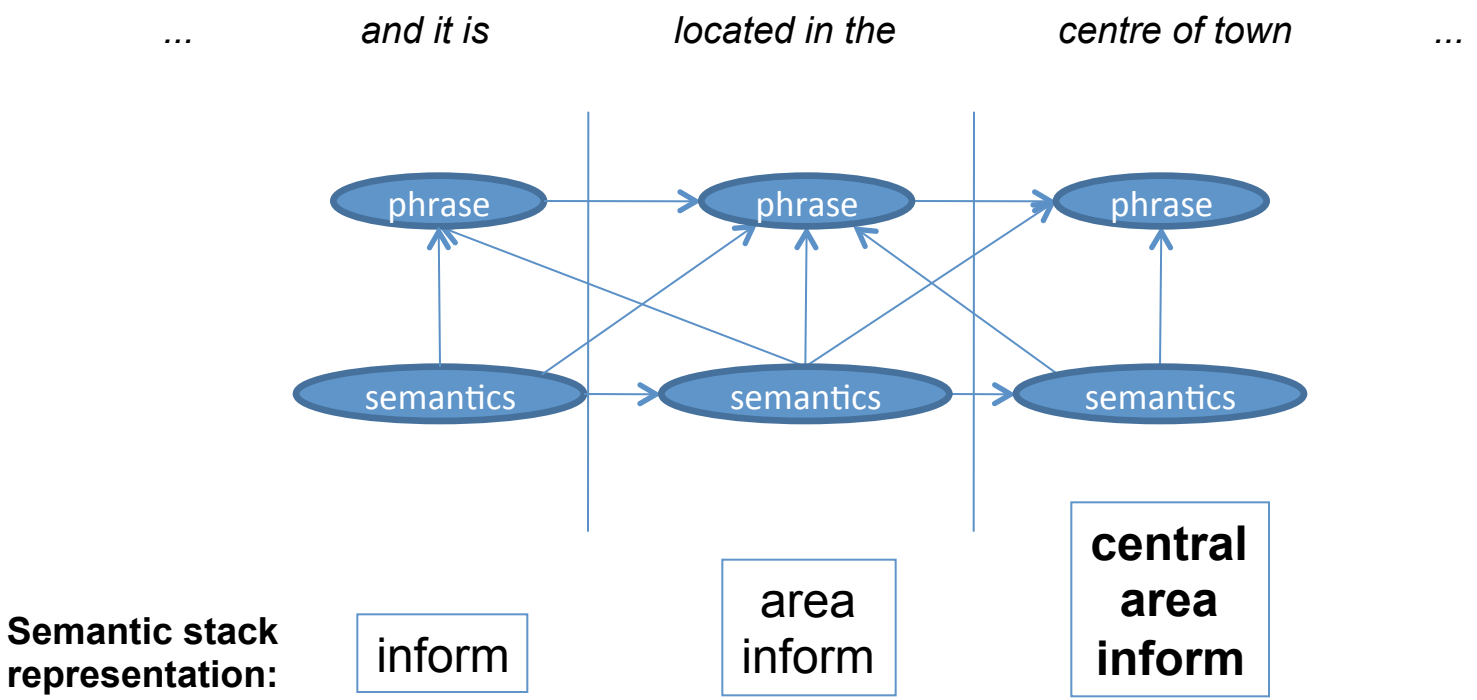
The semantics define the granularity of the model

Observation unit = whatever phrase is associated with a semantic concept

Graphical Models for NLG

Probabilistic model

P(phrase, semantics | semantic context, previous phrases)



Graphical Models for NLG

Training:
Everything is observed, we can just get the maximum likelihood estimates of the probability distributions

Decoding:

Find most likely sequence of phrases given input
e.g. inform(name(Gourmet Burger), none(phone), area(central))
→ *'Gourmet Burger is in the city centre, but there is no information about the phone number'*

The only thing that we know is an *un-ordered set* of mandatory semantic stacks

→ Must search over all possible orderings of stacks as well as over all possible phrase sequences

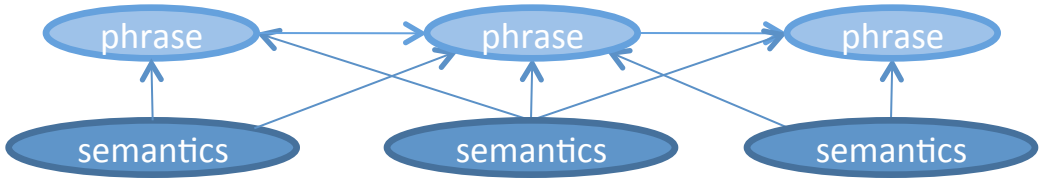
3 Stage FLM Decoding

Decoding semantics and phrases together is not tractable for real time dialogue

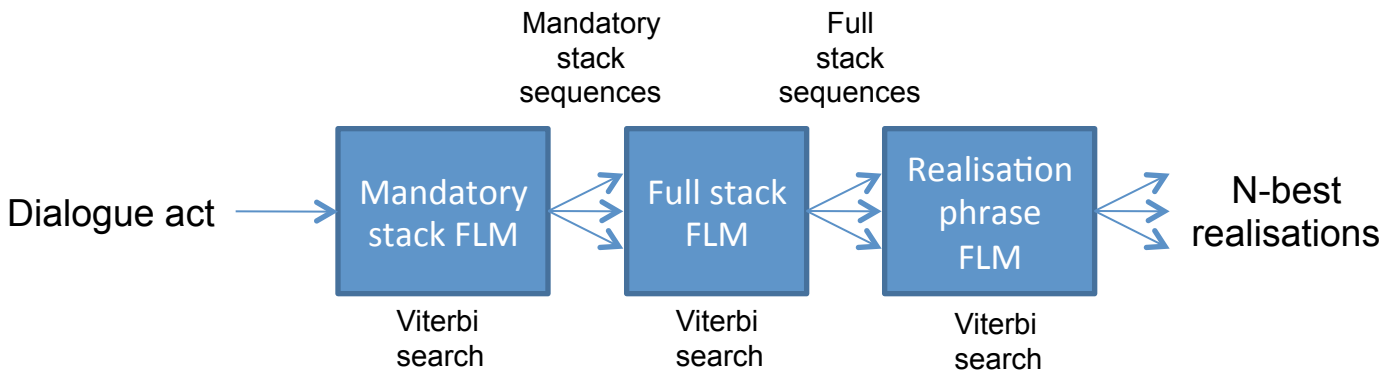
1.Viterbi search over *mandatory* semantic stacks sequence given a trigram semantic model

2.Viterbi search over the *full* set of stacks matching the order of mandatory stacks determined in phase 1

3.Find most likely phrase sequences given full semantic stack sequence (phrase bigram)



3 Stage FLM Decoding

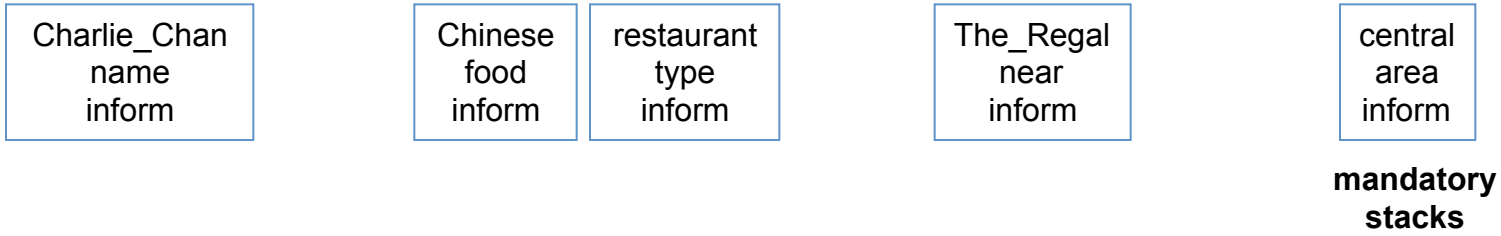


3 Stage FLM Decoding

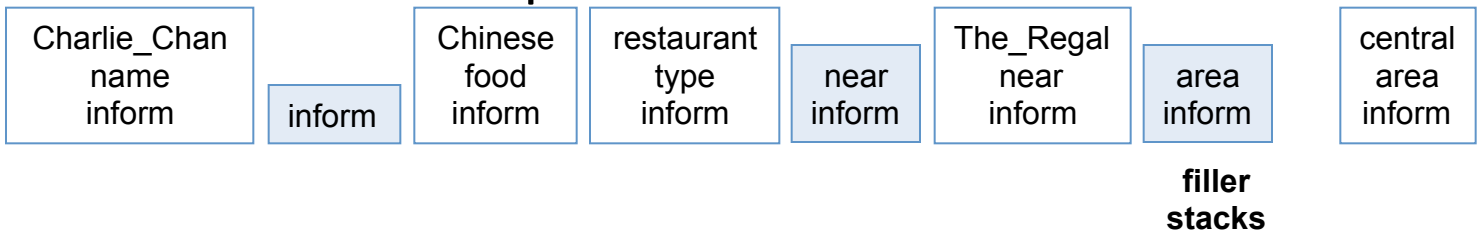
Input dialogue act:

inform(name(Charlie_Chan) type(restaurant) food(Chinese) area(central) near(the_Regal))

Phase 1: mandatory stack order



Phase 2: full stack sequence

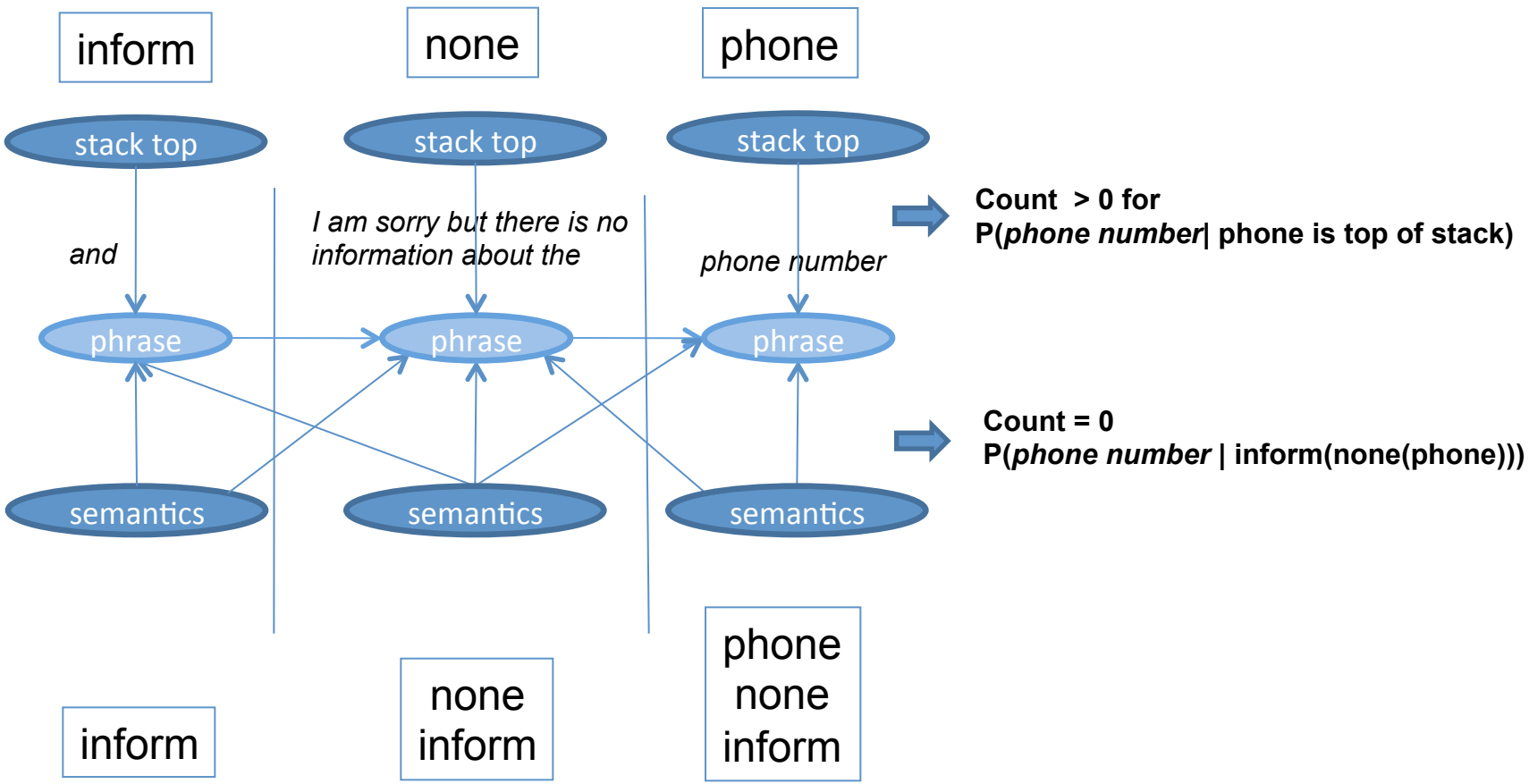


Phase 3: realisation phrase

'Charlie_Chan is_a Chinese restaurant near the_Regal in_the centre_of_town'

Data sparsity issue: handle unseen semantic stacks using factored language models and backoff

- E.g. Count (inform(none(phone))) = 0 during training
  but we have seen *phone* in other context, e.g. inform(phone(0122343))
- Backoff to the top semantic concept (*stacktop* variable)



Small Domain Evaluation

Subset of the Cambridge tourist information system

2 dialogue act types (inform, nomatch)

8 restaurant attributes: name, food, near, pricerange, postcode, phone, address, and area

20,000 simulated dialogues, resulting in 202 unique dialogue acts after replacing proper noun pre-terminal concepts by a generic symbol

NLG Data Collection

Objective 1: train a language generator from data

Objective 2: scalability to new domains

Objective 3: natural paraphrasing capability (e.g. to reduce repetitions during dialogue)

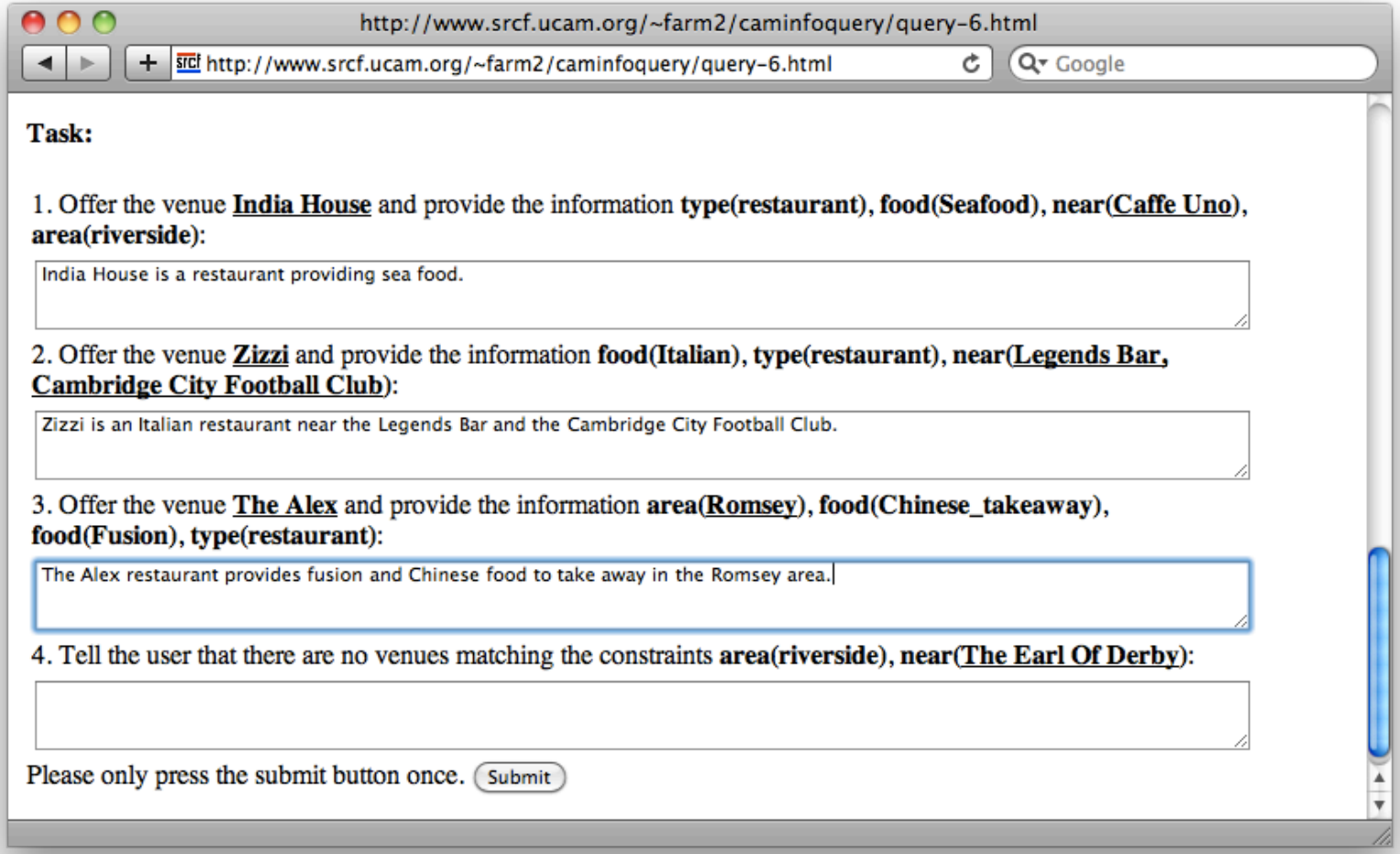
→ Get annotations from a large range of untrained judges (i.e., Mechanical Turks)

Crowd-Sourcing Data Collection for NLG

Issue: how to convey what to generate to *untrained* annotators?

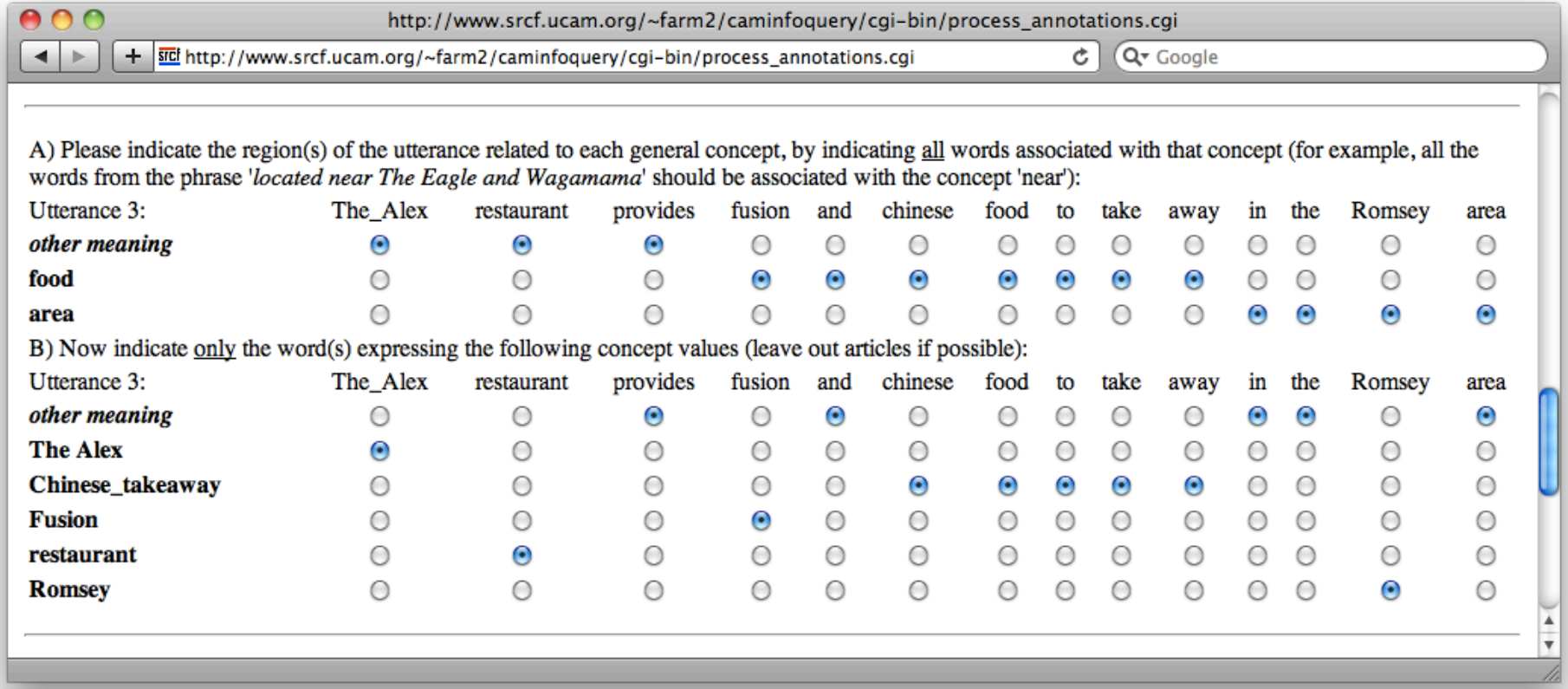
Annotation Phase 1 – Utterance Generation

Annotators produce utterances based on a generic dialogue act type description and a predicate/argument representation



Annotation Phase 2 – Semantic Alignment

Annotators label regions associated with the bottom concepts of each stack, then identify top concepts within each region



Corpus Statistics

42 native speakers of English annotating 404 dialogue acts (two paraphrases per act)

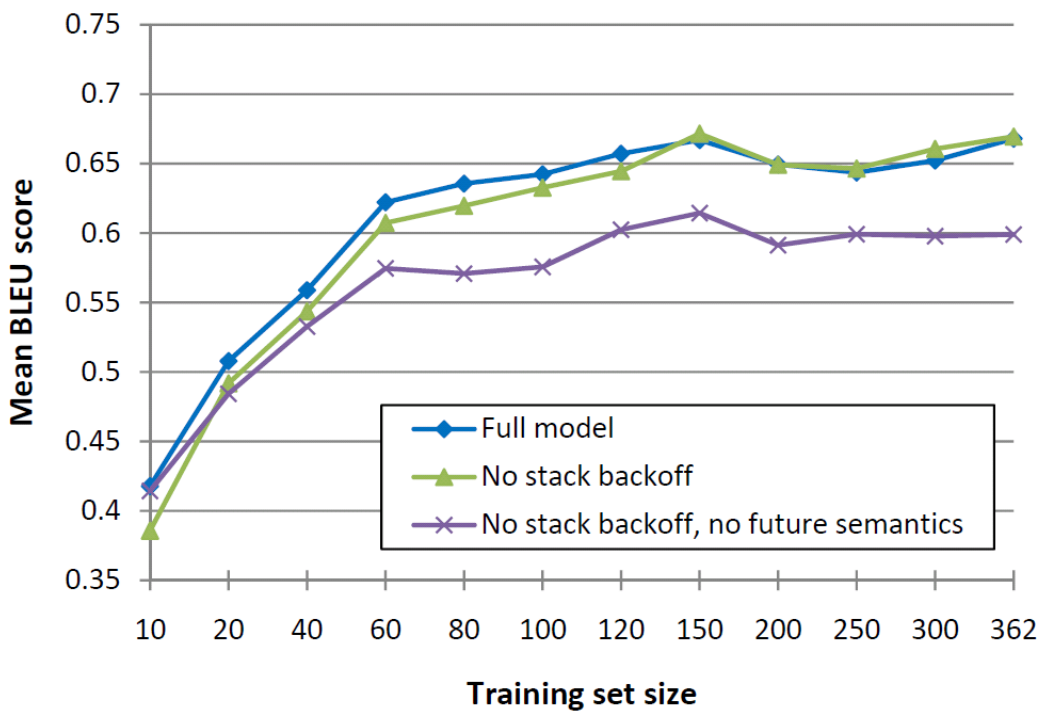
Resulting vocabulary:
52 semantic stacks

109 realisation phrases

Parameter Tuning - BLEU Score Evaluation

N-gram overlap between generated utterance and two paraphrases over a 10 fold cross-validation
→ BLEU score for 20 **unseen** dialogue acts for each fold

Future semantic dependency improves performance



Optimisation of the data collection process

Can we learn faster?

Our model can give us a confidence score for every possible semantic input
= P(most likely phrase sequence|input dialogue act)

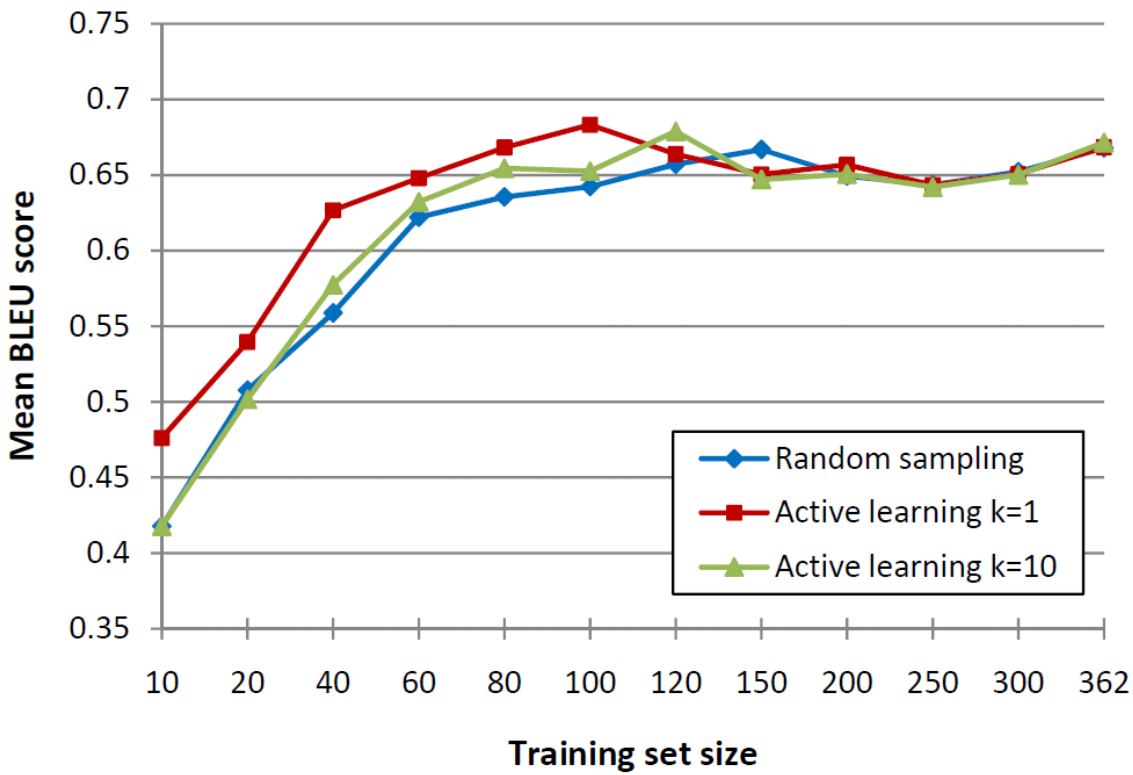
After having seen *n* data points, generate all utterances in the domain, and query annotations for the utterances generated with the lowest confidence

→ **Uncertainty-based active learning**

Parameter Tuning - BLEU Score Evaluation

Active learning by querying the act minimising p(u|act, model) outperforms random sampling

Querying multiple acts simultaneously reduces performance (k=10)



Human Evaluation

Can we produce natural and informative outputs?

Can active learning reduce development time?

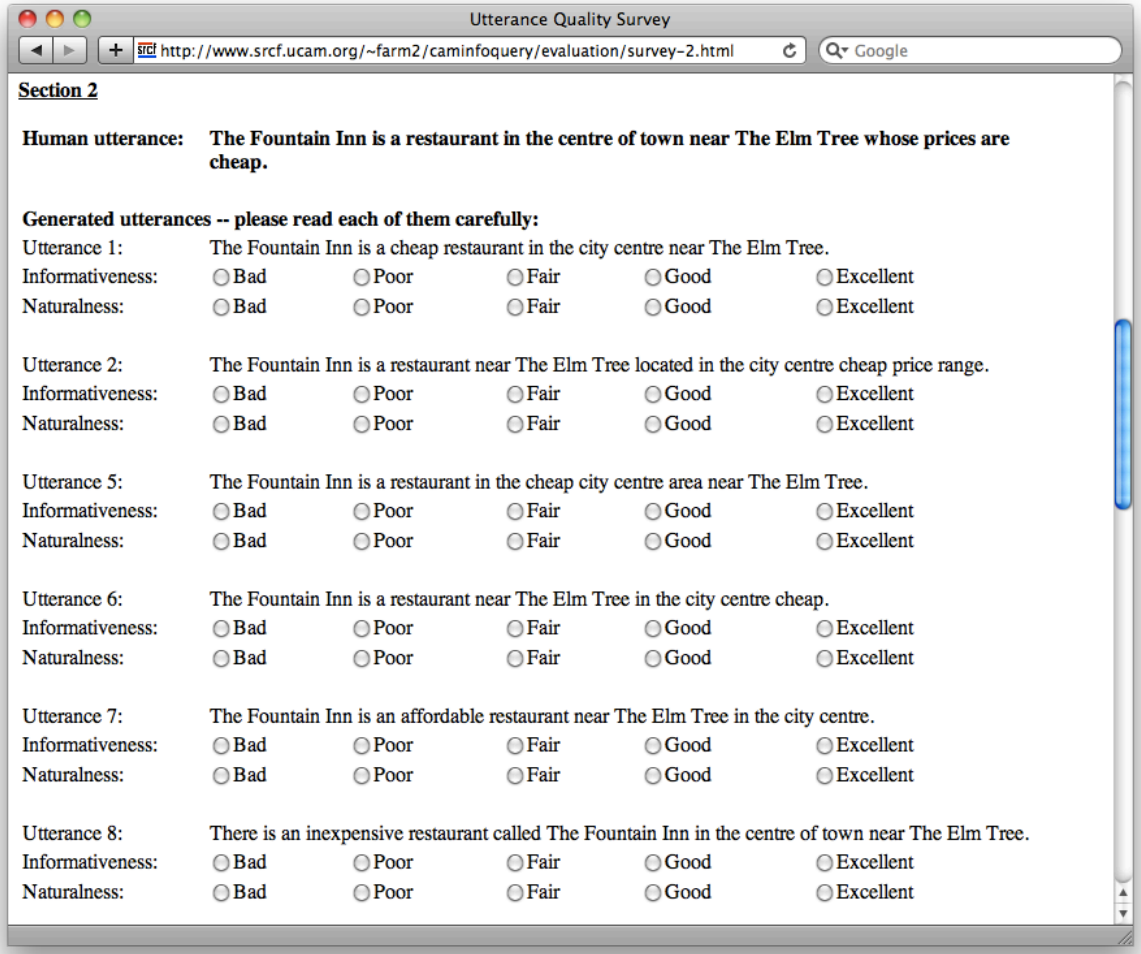
We compare the naturalness and informativeness of utterances generated from ***unseen*** dialogue acts

- Active learning vs. random sampling
- 20, 40, 100, or 362 training utterances

The judges also evaluate a human paraphrase (gold standard)

Human Evaluation

18 native speakers evaluate up to 8 utterances per act for 202 dialogue acts which were not used for training



Utterance Quality Survey

http://www.srcf.ucam.org/~farm2/caminfoquery/evaluation/survey-2.html

Section 2

Human utterance: **The Fountain Inn is a restaurant in the centre of town near The Elm Tree whose prices are cheap.**

Generated utterances -- please read each of them carefully:

Utterance 1: The Fountain Inn is a cheap restaurant in the city centre near The Elm Tree.
Informativeness: Bad Poor Fair Good Excellent
Naturalness: Bad Poor Fair Good Excellent

Utterance 2: The Fountain Inn is a restaurant near The Elm Tree located in the city centre cheap price range.
Informativeness: Bad Poor Fair Good Excellent
Naturalness: Bad Poor Fair Good Excellent

Utterance 5: The Fountain Inn is a restaurant in the cheap city centre area near The Elm Tree.
Informativeness: Bad Poor Fair Good Excellent
Naturalness: Bad Poor Fair Good Excellent

Utterance 6: The Fountain Inn is a restaurant near The Elm Tree in the city centre cheap.
Informativeness: Bad Poor Fair Good Excellent
Naturalness: Bad Poor Fair Good Excellent

Utterance 7: The Fountain Inn is an affordable restaurant near The Elm Tree in the city centre.
Informativeness: Bad Poor Fair Good Excellent
Naturalness: Bad Poor Fair Good Excellent

Utterance 8: There is an inexpensive restaurant called The Fountain Inn in the centre of town near The Elm Tree.
Informativeness: Bad Poor Fair Good Excellent
Naturalness: Bad Poor Fair Good Excellent

Human Evaluation (ACL 2010)

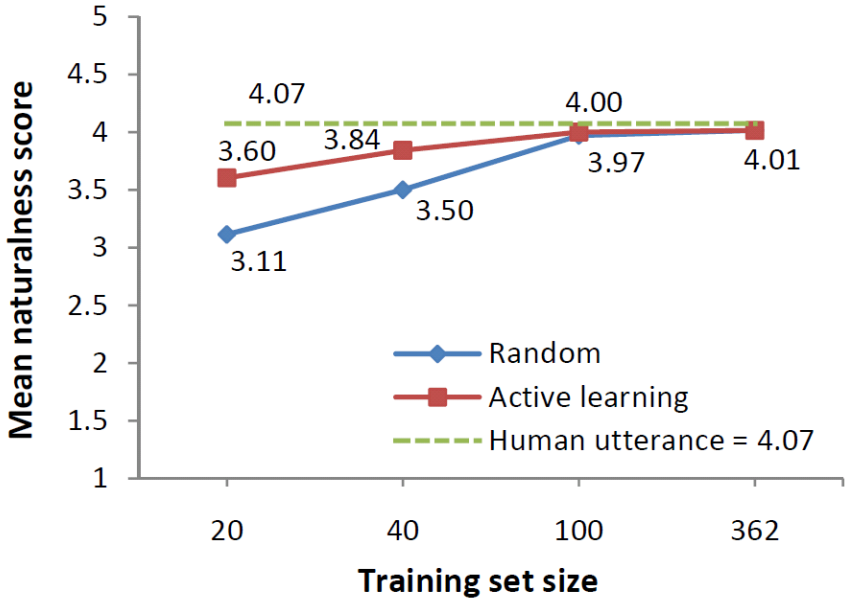
Model trained on the full training set

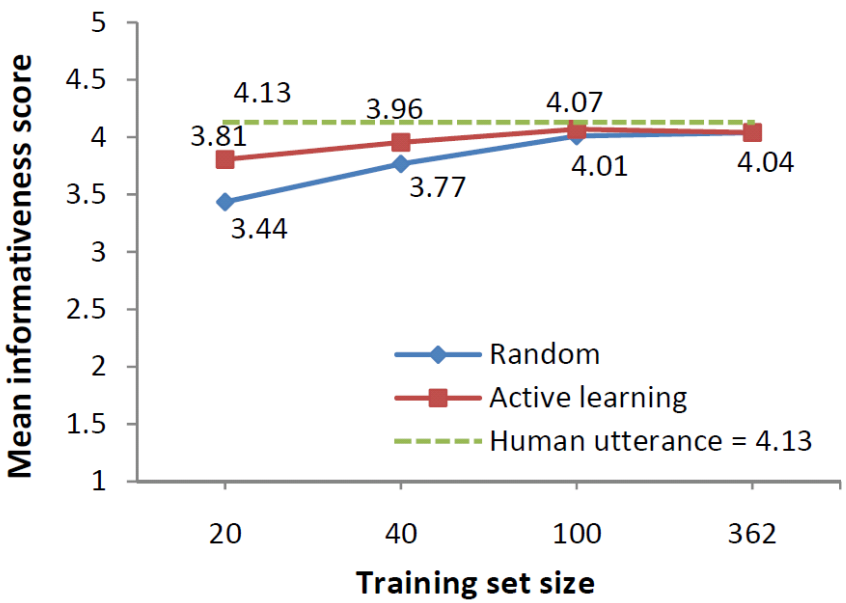
Mean naturalness score: 4.01 out of 5

Human paraphrases: 4.07

Not significantly different (paired t-test with n=202)

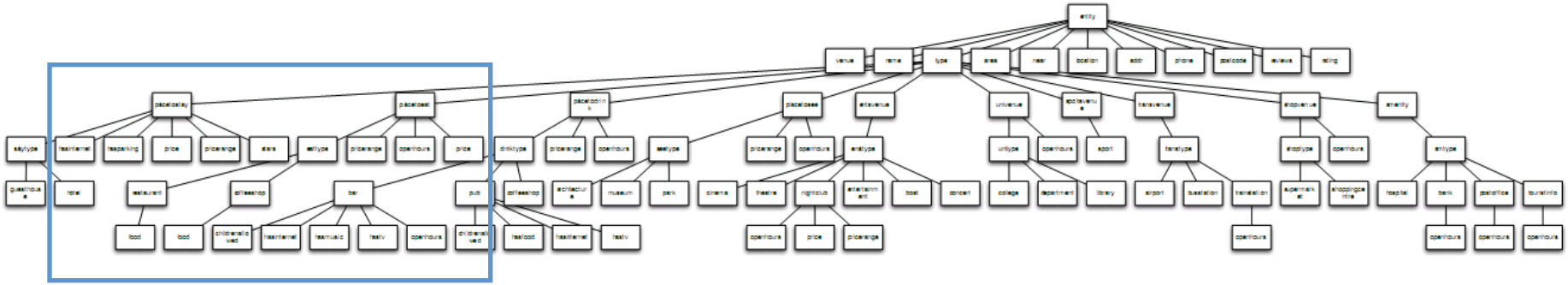
Active learning performs significantly better than random sampling for low training set sizes

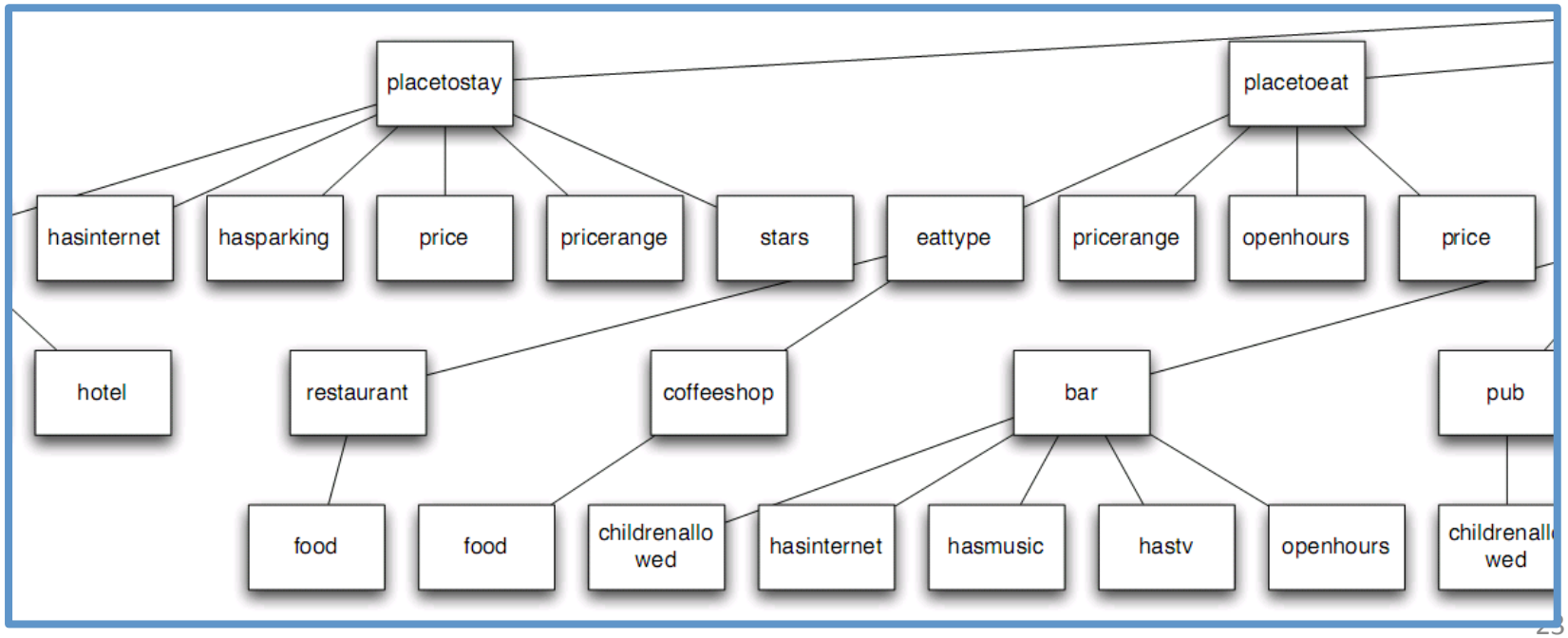




Scaling to Larger Domains under Real-Time Constraints

Cambridge Information System ontology





Scaling to Larger Domains under Real-Time Constraints

Much larger vocabulary

41 types of entities: coffee shops, pubs, supermarket, train station, night clubs, hotels, cinemas, museum, theatres, architectural buildings, sport facilities, etc.

16 types of attributes: price, food type, area, address, phone number, nearby venues, internet connection, parking facilities, children allowed, etc.

More complex utterances

E.g. '*Charlie Chan is a great restaurant in the centre, however it does not have any parking space, and I do not know anything about the pricerange*'

Large real-time decoding constraint (< 0.2 sec)!

Scaling to Larger Domains under Real-Time Constraints

Viterbi decoding performance depends on context size

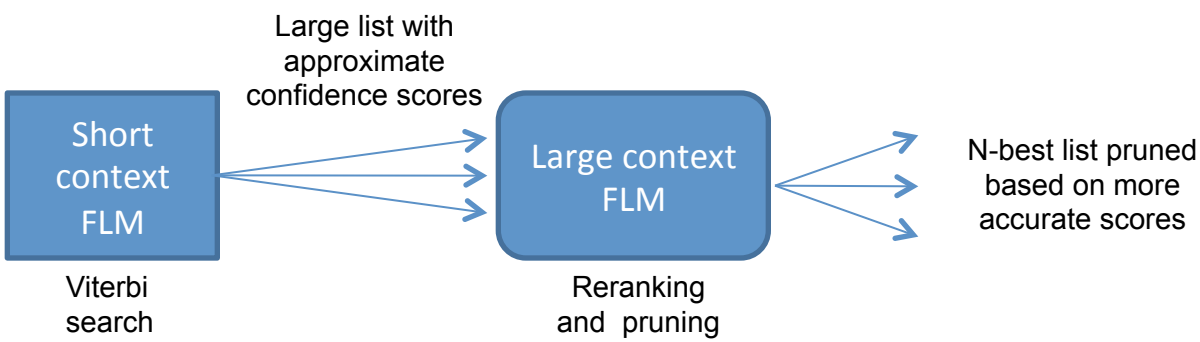
e.g. n-gram model of vocabulary V

state space size $= V^{n-1}$

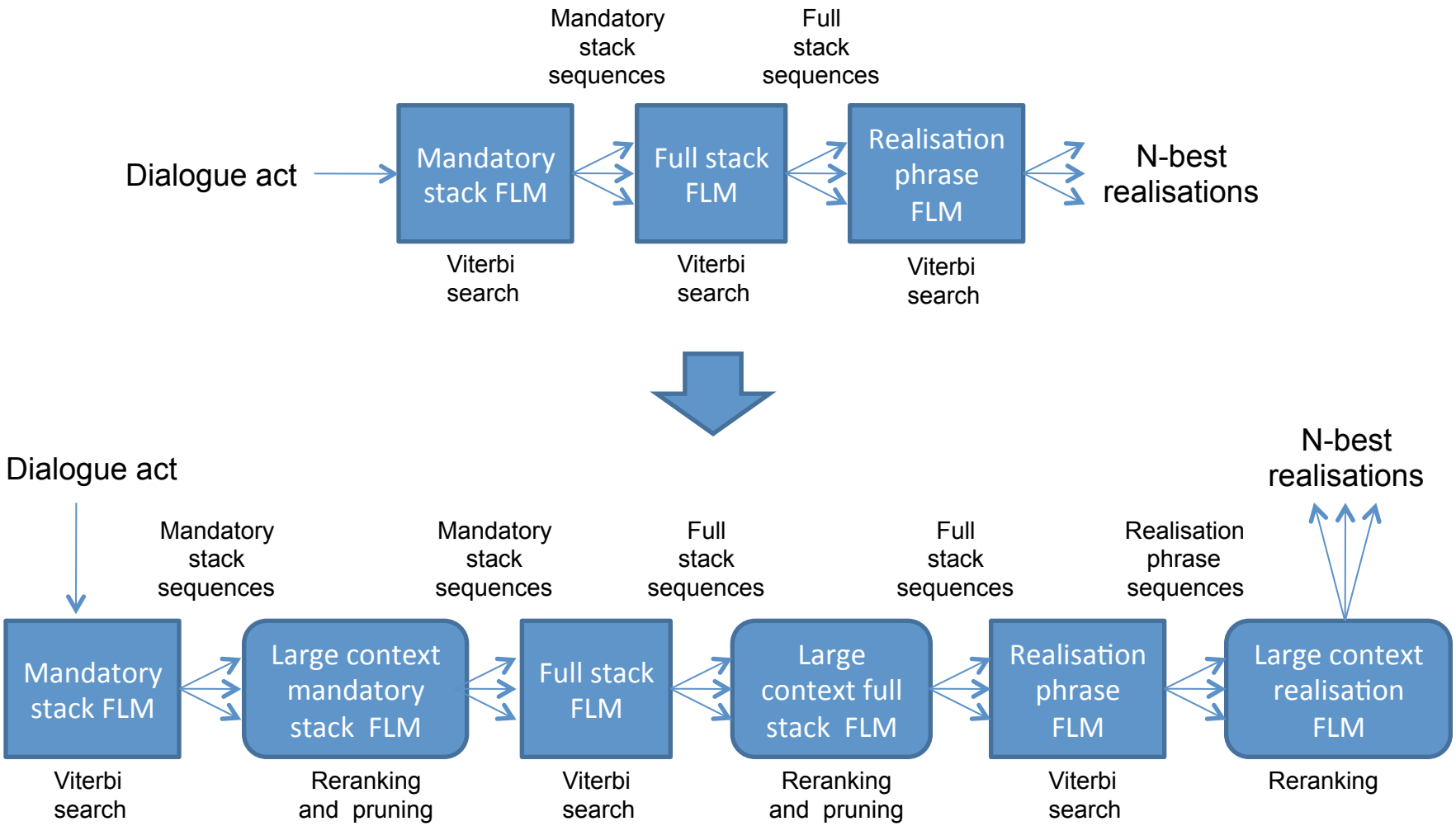
→ Viterbi ~ $O(sequenceLength \,.\, V^n)$

Solution:

1. Costly Viterbi search using limited context models (e.g. bigram)
2. Rerank N-best hypotheses using **large context models** (e.g. 4-gram)



Cascaded NLG using FLMs



Limitations of Factored Language Models

Difficult to find optimal backoff strategy

E.g. shall I backoff to the previous phrase, or to the underlying semantics, or both?

FLMs maximise likelihood of the data rather than more relevant metrics for NLG

- E.g. BLEU, however a single word error can affect the whole utterance
- 0/1 loss more appropriate?

→Discriminative training for *structured prediction*

i.e. output space = all possible phrase sequences

Discriminative training using the Averaged Structured Perceptron (Collins, 2000)

The structured perceptron learns a discriminant function over features of the input and output structure (x, y)

$$\hat{y} = \underset{y \in \mathcal{Y}}{argmax} \ \overrightarrow{w} \cdot \Phi(x, y)$$

The training algorithm iterates through the *N* training examples, updating its weights if the current example *n* is misclassified:

$$\overrightarrow{w} \leftarrow \overrightarrow{w} + \Phi(x_n, y_n) - \Phi(x_n, \hat{y}_n)$$

Each reranking model can be trained using a perceptron, which minimises the training error

Discriminative training using the Averaged Structured Perceptron (Collins, 2000)

Feature vector $\Phi(x_n, y_n)$

E.g. Feature i = *'cuisine'* follows *'Chinese'* AND top semantic = FOOD

Let's assume the Viterbi search generates
'*Would you like some Chinese eat*'

instead of

'*Would you like some Chinese food*'

→ Increase the weight of feature

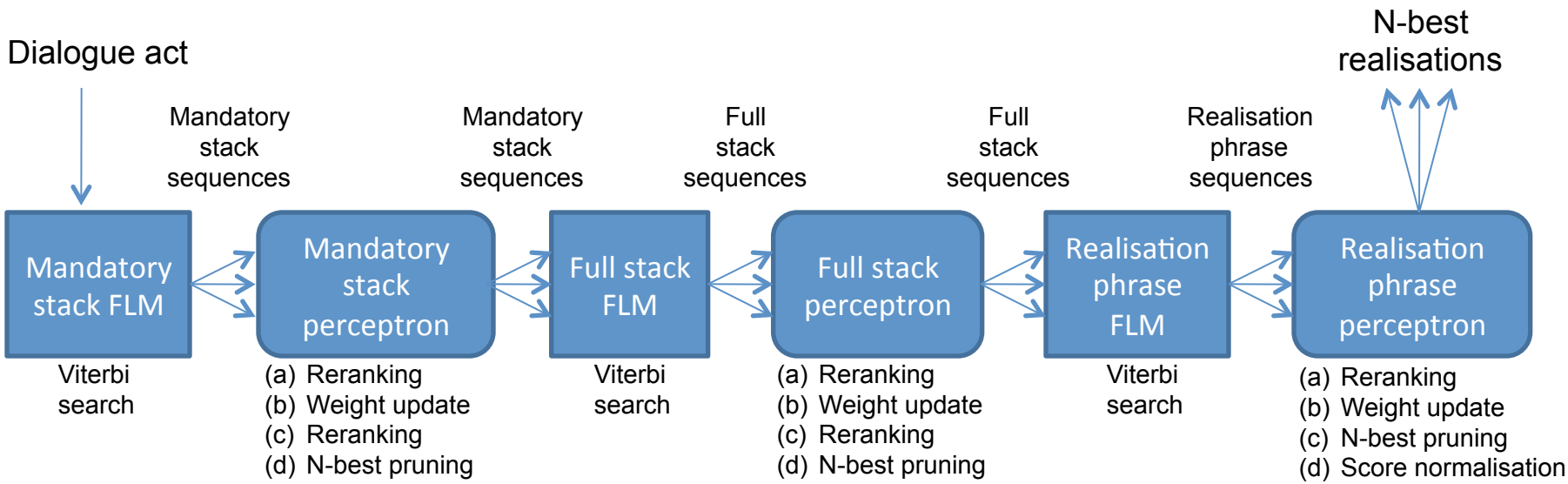
'food' follows *'Chinese'* AND top semantic = FOOD

→ Decrease the weight of feature

'eat' follows *'Chinese'* AND top semantic = FOOD

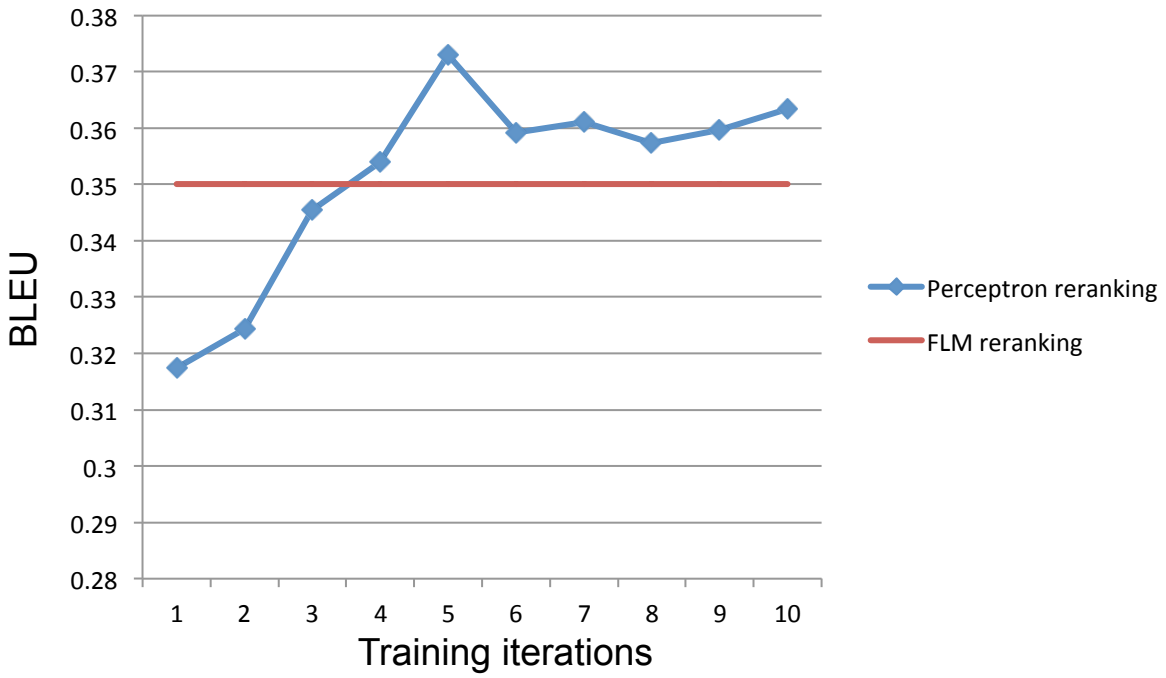
At the next iteration, the dot product $\vec{w} \cdot \Phi(x, y)$ will favour '*Chinese food*' over '*Chinese eat*'

Cascaded NLG using Perceptron Models



BLEU score vs. Perceptron training iterations

Performs slightly better than FLM reranking on unseen data



But most importantly:

It only requires a dot product at run time!

Data-Driven NLG for Improving Dialogue Naturalness

Human-machine dialogue can be repetitive and monotonous

Can we use statistical models to improve dialogue naturalness by using the output N-best list for paraphrasing?

N-Best List Output

Dialogue act: nomatch(area(Arbury) near(the Portland Arms))

-9.78180: I am sorry but i know of no venues near the Portland Arms in the Arbury area

-9.78180: I am sorry but there are no venues near the Portland Arms in the Arbury area

-9.78180: I am sorry but there are no venues matching your constraints near the Portland Arms in the Arbury area

-9.78180: I regret to tell you that there are no places i know of near the Portland Arms in the Arbury area

-9.78180: there are no places you are looking for near the Portland Arms in the Arbury area

-10.0382: there are no venues in the Arbury area near the Portland Arms

-10.6850: I am sorry there is no information about a venue near the Portland Arms in the Arbury area

-10.8101: I am sorry but there is no information about a venue near the Portland Arms in the Arbury area

-10.8100: I am sorry there is no information matching the constraints near the Portland Arms in the Arbury area

Ongoing Work

Sample from the output N-best list rather than picking the 1st best utterance

Trade-off between paraphrasing capability and naturalness

How can we evaluate this?

Ideally, get user perception after interacting with the system

In practice,

- We re-synthesise the system utterances of existing dialogues
- Ask observers to rate the naturalness of the system's output over the dialogue

→ *Can we improve the user perception of the system?*

Conclusion

Dynamic Bayesian Networks can be used to train a generator with *no handcrafting* beyond the definition of the semantics

Untrained annotators can produce aligned data to reduce development time

Active learning can be used to guide the data collection process

Structured Perceptron Reranking can improve real-time decoding performance

Data-driven methods and crowdsourcing offer the potential for more natural conversational interfaces by implicitly modelling individual differences in linguistic variation!

Thank you

Semantically-aligned corpus: **mi.eng.cam.ac.uk/~farm2**

Any question?