NPFL099 Statistical Dialogue Systems
4. Language Understanding

http://ufal.cz/npfl099

Ondřej Dušek & **Vojtěch Hudeček**

20. 10. 2020

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

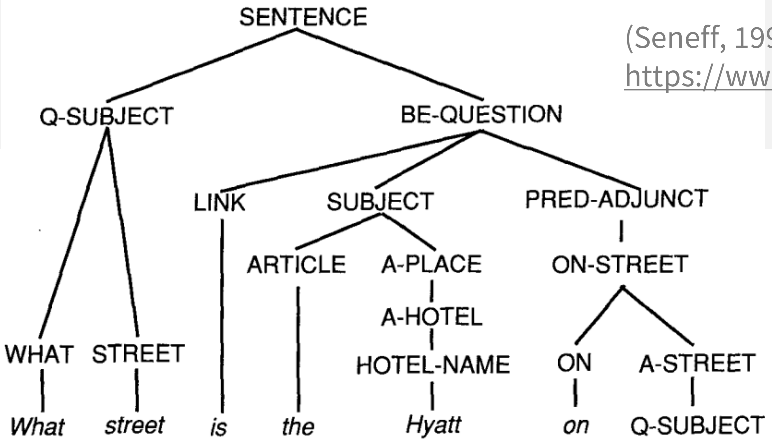
Natural Language Understanding

- **words → meaning**
  - whatever "meaning" is – can be different tasks
  - typically structured, explicit representation
- alternative names/close tasks:
  - **spoken language understanding**
  - **semantic decoding/parsing**
- integral part of dialogue systems, also explored elsewhere
  - stand-alone semantic parsers
  - other applications:
    - human-robot interaction
    - question answering
    - machine translation (not so much nowadays)

NLU Challenges

- non-grammaticality *find something cheap for kids should be allowed*
- disfluencies
  - hesitations – pauses, fillers, repetitions *uhm I want something in the west the west part of town*
  - fragments *uhm I'm looking for a cheap*
  - self-repairs (~6%!) *uhm find something uhm something cheap no I mean moderate*
- ASR errors *I'm looking for a for a chip Chinese rest or rant*
- synonymy *Chinese city centre*
  I've been wondering if you could find me a restaurant that has Chinese food close to the city centre please
- out-of-domain utterances *oh yeah I've heard about that place my son was there last month*

Semantic representations



(Seneff, 1992)
https://www.aclweb.org/anthology/J92-1004

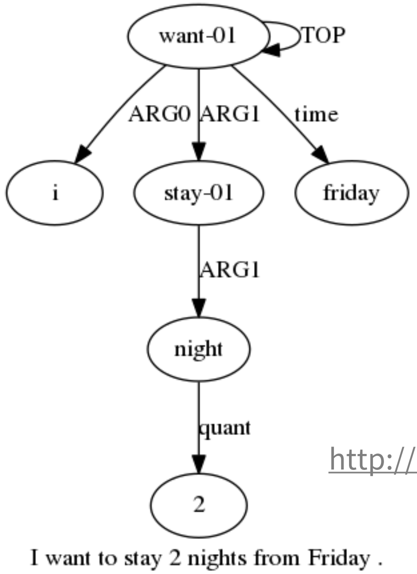
- syntax/semantic **trees**
  - typical for standalone semantic parsing
  - different variations
- **frames**
  - technically also trees, but smaller, more abstract
  - (mostly older) DSs, some standalone parsers
- **graphs** (AMR)
  - trees + co-reference
    (e.g. pronouns referring to the same object)
- **dialogue acts** = intent + slots & values
  - flat – no hierarchy
  - most DSs nowadays

inform(date=Friday, stay="2 nights")

oui l'hôtel don't le prix ne dépasse pas cent dix euros

response:	oui			
refLink:	co-ref.			
	singular			
BDObject:	hotel			
		room		
		payment:	amount	
			comparative:	less
			integer:	110
			unit:	euro

(Bonneau-Maynard et al., 2005)
https://www.isca-speech.org/archive/interspeech_2005/i05_3457.html



http://cohort.inf.ed.ac.uk/amreager.html

Handling ASR noise

- ASR produces **multiple hypotheses**
- Combine & get resulting NLU hypotheses
  - NLU: $p(\text{DA}|\text{text})$
  - ASR: $p(\text{text}|\text{audio})$
  - we want $p(\text{DA}|\text{audio})$
- Easiest: **sum it up**

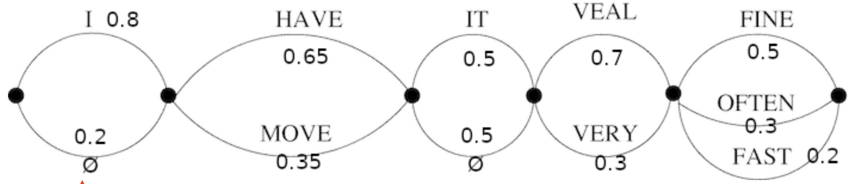
$$p(\text{DA}|\text{audio}) = \sum_{\text{texts}} P(\text{DA}|\text{text}) P(\text{text}|\text{audio})$$

```
0.33 – I am looking for a bar
0.26 – I am looking for the bar
0.11 – I am looking for a car
0.09 – I am looking for the car

0.59 – inform(task=find, venue=bar)
0.20 – null()
```

(from Filip Jurčíček's slides)

- Alternative: **confusion nets** with weighted words
  - a more concise way of showing the same thing

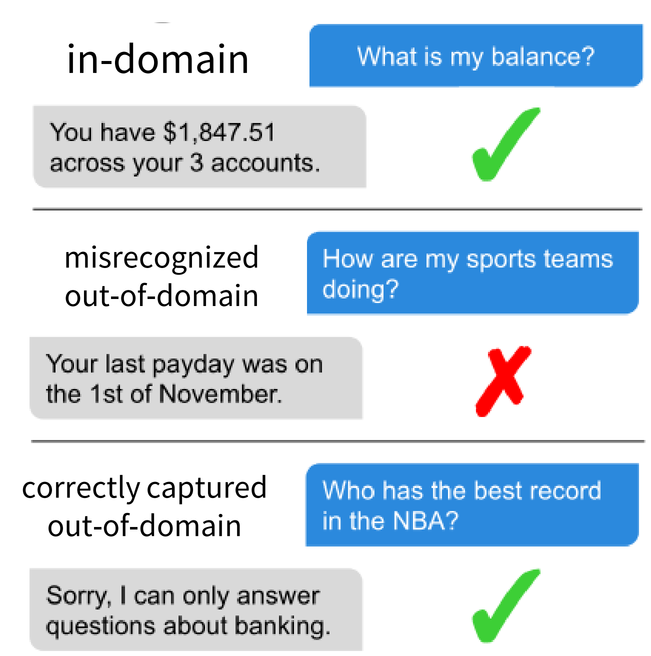


(skiping a word)

left-to-right: multiply probabilities

Handling out-of-domain queries

- Handcrafted: **no pattern matches** → out-of-domain
- Datasets – rarely taken into account!
- **Low confidence** on any intent → out-of-domain?
  - might work, but likely to fail (no explicit training for this)
- Out-of-domain data + **specific OOD intent**
  - adding OOD from a different dataset
    - problem: “out-of-domain” should be broad, not just some different domain
  - collecting out-of-domain data specifically
    - worker errors for in-domain
    - replies to specifically chosen irrelevant queries
  - always need to ensure that they don’t match any intent randomly
  - not so many instances needed (expected to be rare)



(Larson et al., 2019)
http://arxiv.org/abs/1909.02027

NLU as classification

- using DAs – treating them as a **set of semantic concepts**
  - concepts:
    - intent
    - slot-value pair
  - binary classification: is concept Y contained in utterance X?
  - independent for each concept
- consistency problems
  - conflicting intents (e.g. *affirm* + *negate*)
  - conflicting values (e.g. *kids-allowed=yes* + *kids-allowed=no*)
  - need to be solved externally, e.g. based on classifier confidence

NER + delexicalization

- Approach:

1) **identify** slot values/named entities
2) **delexicalize** = replace them
   with placeholders (indicating entity type)
   - or add the NE tags as more features for classification

- generally needed for NLU as classification
  - otherwise in-domain data is too sparse
  - this can vastly reduce the number of concepts to classify & classifiers
- NER is a problem on its own
  - but general-domain NER tools may need to be adapted
  - in-domain gazetteers, in-domain training data

What is the phone number for Golden Dragon?
What is the phone number for <restaurant-name>?

I'm looking for a Japanese restaurant in Notting Hill.
I'm looking for a <food> restaurant in <area>.

I need to leave after 12:00.
I need to leave after <time>.
*leave_at -> **leave_at***
*arrive_by -> **none***

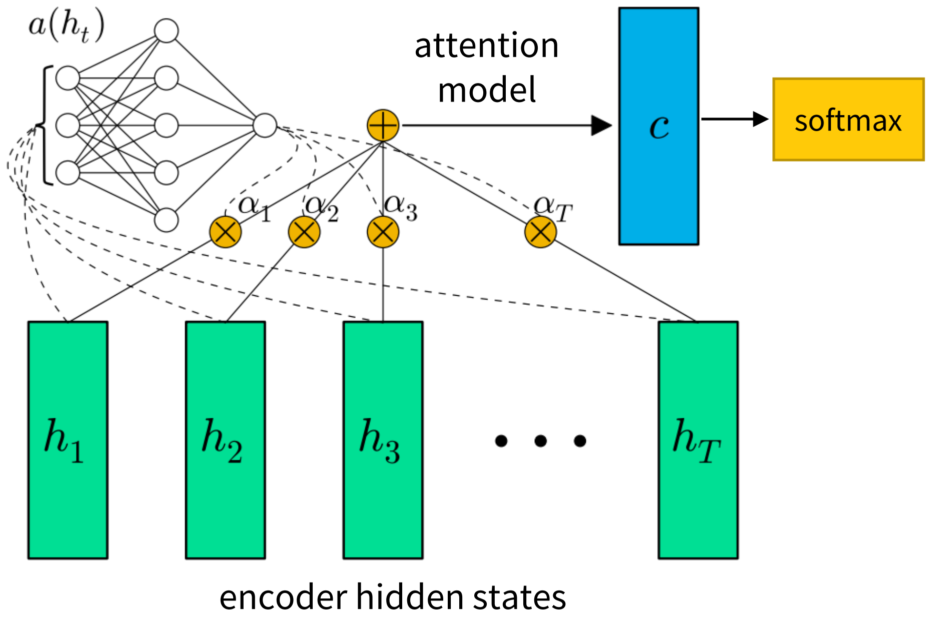
Both can be <time>

NLU Classifier models

- note that data is usually scarce!
- **handcrafted / rules**
  - simple mapping: word/n-gram/regex match → concept
  - can work really well for a limited domain
  - no training data, no retraining needed (tweaking on the go)
- **linear classifiers**
  - logistic regression, SVM…
  - need handcrafted features
- **neural nets** (=our main focus today)

NN neural classifiers

- **intent** = **multi-class** (softmax)
- **slot** tagging = set of **binary classifiers** (logistic loss)
- using word embeddings (task-specific or pretrained)
  - no need for handcrafted features
  - still needs delexicalization (otherwise data too sparse)
- different architectures possible
  - bag-of-words feed-forward NN
  - RNN / CNN encoders + classification layers
  - attention-based



(Raffel & Ellis, 2016)
https://colinraffel.com/publications/iclr2016feed.pdf

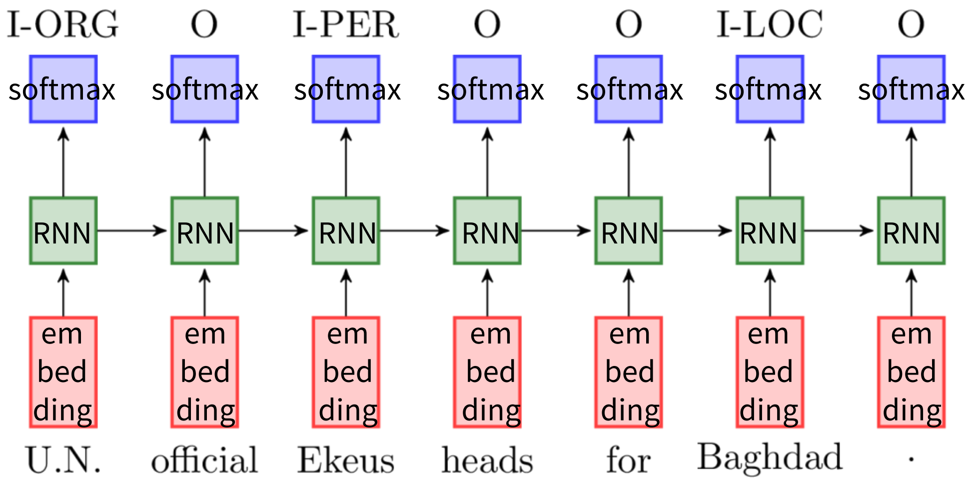
Slot filling as sequence tagging

- get slot values directly – no need for delexicalization
  - each word classified
  - classes = slots & **IOB format** (inside-outside-beginning)
  - slot values taken from the text
    (where a slot is tagged)
  - NER-like approach
- rules + classifiers still work
  - keywords/regexes found at specific position
  - apply classifier to each word in the sentence left-to-right
- linear classifiers are still an option

I	*need*	*a*	*flight*	*from*	*Boston*	*to*	*New*	*York*	*tomorrow*
O	**O**	**O**	**O**	**O**	**B-dept**	**O**	**B-arr**	**I-arr**	**B-date**

Neural sequence tagging

- Basic neural architecture:
  RNN (LSTM/GRU) → softmax over hidden states
  - + some different model for intents (such as classification)
- Sequence tagging problem: overall consistency
  - slots found elsewhere in the sentence might influence what's classified now
  - may suffer from **label bias**
    - trained on gold data – single RNN step only
    - during inference, cell state is influenced by previous steps – danger of cascading errors
  - solution: **structured/sequence prediction**
    – conditional random fields (CRF)
    - can run CRF over NN outputs

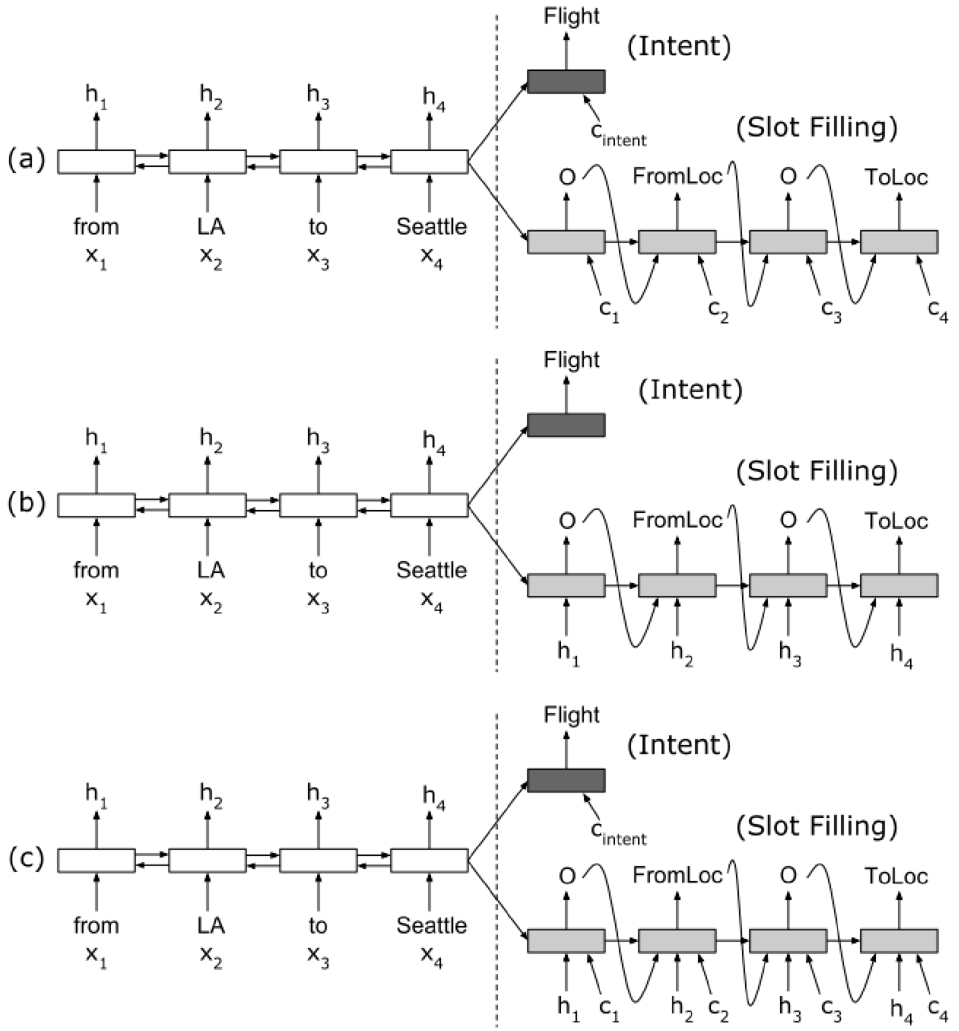


https://www.depends-on-the-definition.com/guide-sequence-tagging-neural-networks-python/

Joint Intent & Slots Model

(Liu & Lane, 2016)
http://arxiv.org/abs/1609.01454

- Same network for both tasks
- Bidirectional encoder
  - 2 encoders: left-to-right, right-to-left
  - “see everything before you start tagging”
- Decoder – tag word-by-word, inputs:
  - attention
  - input encoder hidden states (“aligned inputs”)
  - both
- Intent classification:
  softmax over last encoder state
  - + specific intent context vector c_{intent} (attention)



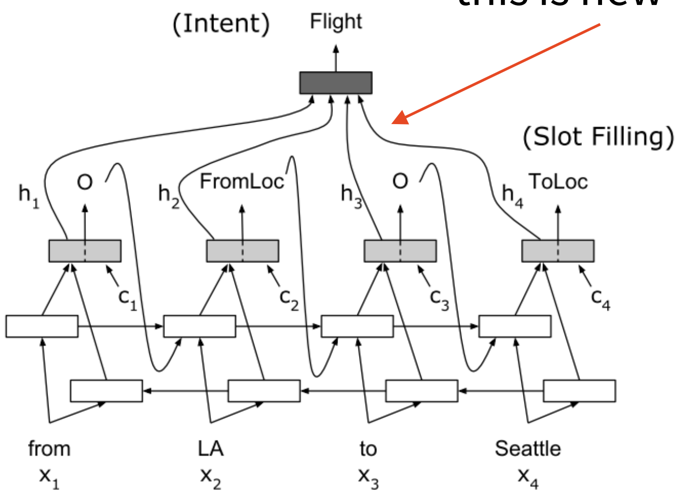
NN for Joint Intent & Slots

(Liu & Lane, 2016)
http://arxiv.org/abs/1609.01454

- Extended version:
  use slot tagging results in intent classification
  - Bidi encoder
  - Slots decoder with encoder states & attention
  - Intent decoder
    – attention over slots decoder states
- Training for both intent & slot detection improves results on ATIS flights data
  - this is multi-task training ☺
  - intent error lower (2% → 1.5%)
  - slot filling slightly better (F1 95.7% → 95.9%)

5k instances
17 intents
~100 slots

this is new



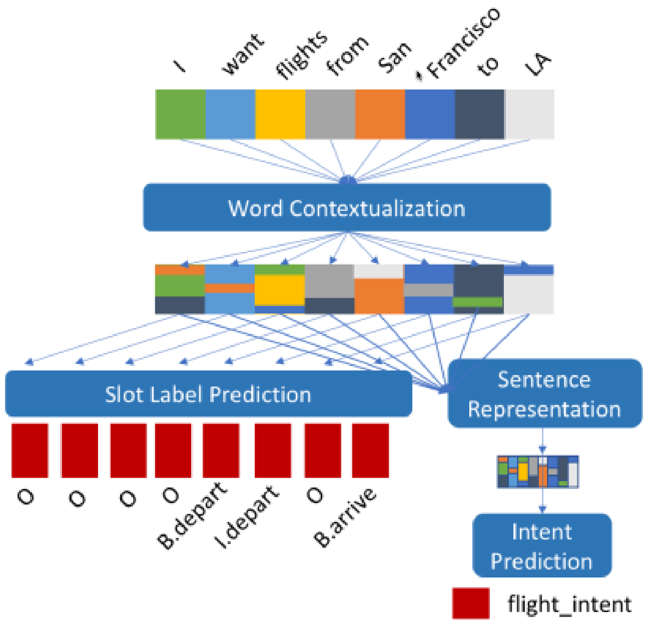
same as (c)
on previous slide

- Variant: treat **intent detection as slot tagging**
  - append <EOS> token & tag it with intent

(Hakkani-Tür et al, 2016)
https://doi.org/10.21437/Interspeech.2016-402

Joint intents & slots with contextual embeddings

(Gupta et al., 2019)
http://arxiv.org/abs/1903.08268



- shared "word contextualization"
  - feed-forward – $\sum$ word + trained position embeddings
  - CNNs
  - (Transformer-style) attention with relative position
    - trained relative position embeddings
      instead of Transformer fixed absolute position embedding
  - LSTM
- task-specific network parts
  - intent: weighted sum of contextualized embeddings + softmax
  - slots tagging:
    - independent – non-recurrent, depend only on current embedding: $P(l_i|\mathbf{h}_i)$
    - label-recurrent – depend on past labels & current embedding: $P(l_i|l_{1,\ldots i-1}, \mathbf{h}_i)$
      - faster than word-recurrent

Joint intents & slots w/context embeddings

- CNN > LSTM > attention > feed-forward
  - CNNs are also faster than anything other than FF
- label-recurrent models mostly better than independent
  - except intent classification (non-recurrent task) on 1 dataset

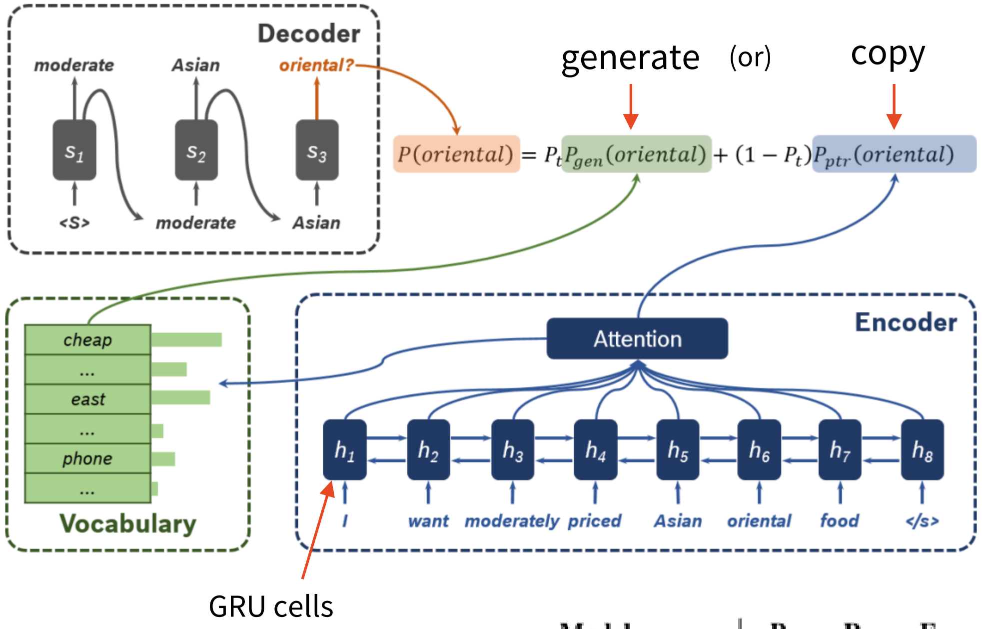
Model	label recurrent	intent classif. accuracy		slot labelling F1		Inference ms/utterance	Epochs to converge	s/epoch	# params
		Snips	ATIS	Snips	ATIS				
FEED-FORWARD	No	**98.56**	97.14	53.59	69.68	0.61	48	1.82	17k
FEED-FORWARD	Yes	98.54	**97.46**	**75.35**	**88.72**	1.82	83	2.52	19k
CNN, 5KERNEL, 1L	No	98.56	98.40	85.88	94.11	0.82	23	1.90	42k
CNN, 5KERNEL, 3L	No	99.04	**98.42**	92.21	96.68	1.37	55	2.16	91k
CNN, 3KERNEL, 4L	No	98.81	98.32	91.65	96.75	1.28	57	2.29	76k
CNN, 5KERNEL, 1L	Yes	98.85	98.36	93.12	96.39	2.13	51	2.77	43k
CNN, 5KERNEL, 3L	Yes	**99.10**	98.36	**94.22**	**96.95**	2.68	59	3.34	93k
CNN, 3KERNEL, 4L	Yes	98.96	98.32	93.71	**96.95**	2.60	53	3.43	78k
ATTN, 1HEAD, 1L, NO-POS	No	98.50	97.51	53.61	69.31	1.95	25	1.94	22k
ATTN, 1HEAD, 1L	No	98.53	97.74	75.55	93.22	4.75	117	4.34	23k
ATTN, 1HEAD, 3L	No	**98.74**	98.10	81.51	94.07	7.68	160	4.32	33k
ATTN, 2HEAD, 3L	No	98.31	98.10	83.02	94.61	7.86	79	4.87	47k
ATTN, 1HEAD, 1L, NO POS	Yes	98.63	97.68	74.94	88.60	3.24	60	2.66	24k
ATTN, 1HEAD, 1L	Yes	98.61	98.00	86.72	94.53	6.12	89	5.53	24k
ATTN, 1HEAD, 3L	Yes	98.51	**98.26**	88.04	94.99	9.03	109	6.06	34k
ATTN, 2HEAD, 3L	Yes	98.48	**98.26**	**89.31**	**95.86**	9.17	93	6.54	49k
LSTM, 1L	No	**98.82**	98.34	91.83	97.28	2.65	45	2.91	47k
LSTM, 2L	No	98.77	98.20	93.10	97.36	4.72	58	5.09	77k
LSTM, 1L	Yes	98.68	**98.36**	93.83	**97.37**	3.98	54	4.62	49k
LSTM, 2L	Yes	98.71	98.30	**93.88**	97.28	6.03	69	6.82	79k

(Gupta et al., 2019)
http://arxiv.org/abs/1903.08268

Seq2seq-based NLU

(Zhao & Feng, 2018)
https://www.aclweb.org/anthology/P18-2068/

- seq2seq with **copy mechanism = pointer-generator net**
  - normal **seq2seq** with attention – generate output tokens (softmax over vocabulary)
  - **pointer net**: select tokens from input (attention over input tokens)
  - prediction = **weighted combination** of
- can work with out-of-vocabulary
  - e.g. previously unseen restaurant names
  - (but IOB tagging can, too)
- generating slots/values + intent
  - it's not slot tagging (doesn't need alignment)
    - **works for slots expressed implicitly or not as consecutive phrases**
  - treats intent as another slot to generate



Model	P	R	F
CNN	**93.5**	78.5	85.3
Seq2Seq w/ attention	87.5	82.7	85.0
Our model	89.0	**82.8**	**85.8**

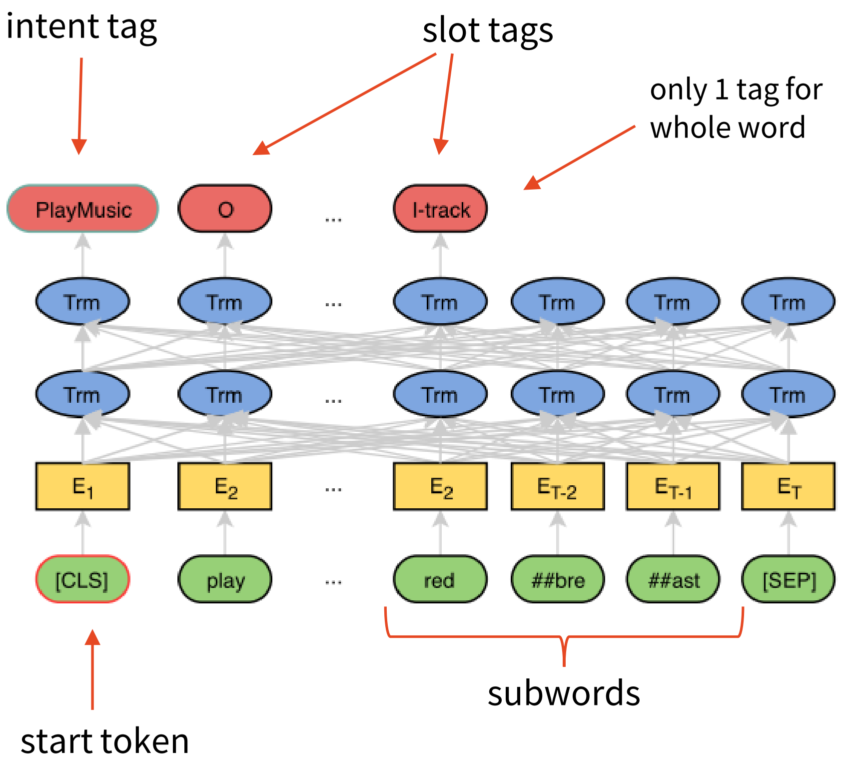
DSTC2 results

Can I bring my kids along to this restaurant? confirm(kids_friendly=yes)
I want a Chinese place with a takeaway option. inform(food=Chinese_takeaway)

BERT-based NLU

(Chen et al., 2019)
http://arxiv.org/abs/1902.10909

- slot tagging on top of pretrained BERT
  - standard **IOB approach**
  - just feed final hidden layers to **softmax over tags**
    - classify only at 1st subword in case of split words (don't want tag changes mid-word)
- special start token tagged with intent
- optional CRF on top of the tagger
  - for global sequence optimization



slightly different numbers, most probably a reimplementation

Models	Snips Intent	Snips Slot	Snips Sent	ATIS Intent	ATIS Slot	ATIS Sent
RNN-LSTM (Hakkani-Tür et al., 2016)	96.9	87.3	73.2	92.6	94.3	80.7
Atten.-BiRNN (Liu and Lane, 2016)	96.7	87.8	74.1	91.1	94.2	78.9
Slot-Gated (Goo et al., 2018)	97.0	88.8	75.5	94.1	95.2	82.6
Joint BERT	**98.6**	**97.0**	**92.8**	97.5	**96.1**	88.2
Joint BERT + CRF	98.4	96.7	92.6	**97.9**	96.0	**88.6**

% completely correct sentences

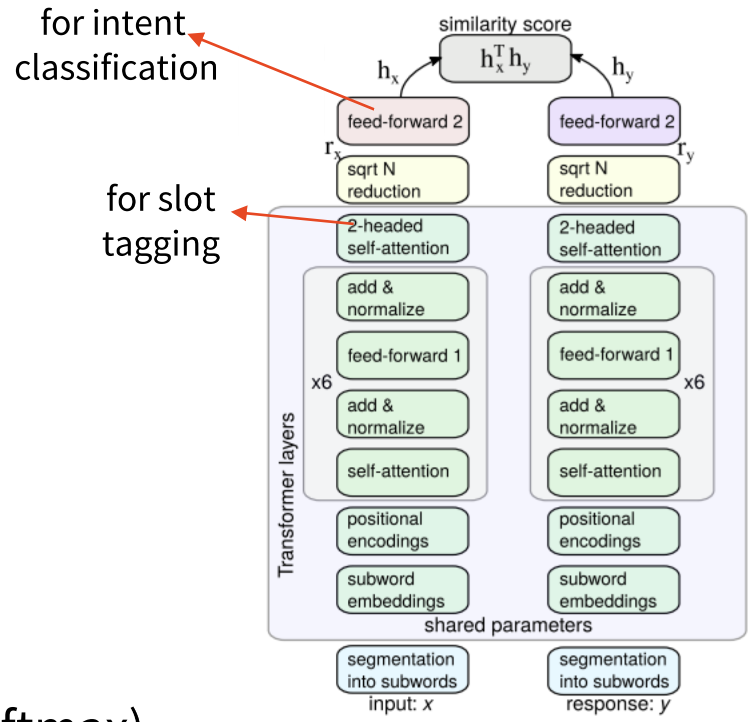
accuracy

F1

Dialogue Pretrained Models

(Henderson et al., 2020)
http://arxiv.org/abs/1911.03688

- Pretraining on dialogue tasks can do better (& smaller) than BERT
  - ConveRT: Transformer-based **dual encoder**
    - 2 Transformer encoders: context + response
      - optionally 3rd encoder with more context (concatenated turns)
    - feed forward + cosine similarity on top
  - training objective: **response selection**
    - response that actually happened = 1
    - random response from another dialogue = 0
  - trained on a large dialogue dataset (Reddit)



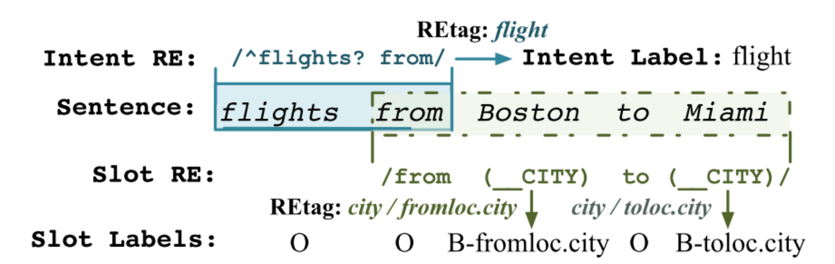
- can be used as a base to train models for:
  - **slot tagging** (top self-attention layer → CNN → CRF)
  - **intent classification** (top feed-forward → more feed-forward → softmax)
  - Transformer layers are fixed, not fine-tuned
  - works well for little training data (**few-shot**)

(Coope et al., 2020)
https://www.aclweb.org/anthology/2020.acl-main.11

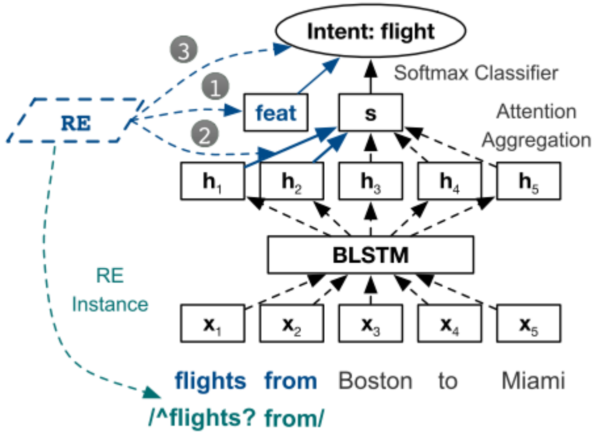
(Casanueva et al., 2020)
https://www.aclweb.org/anthology/2020.nlp4convai-1.5

Regular Expressions & NNs for NLU

(Luo et al., 2018) http://arxiv.org/abs/1805.05588

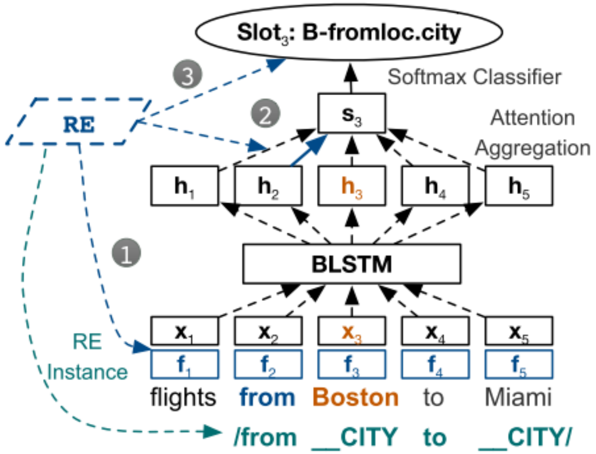


- Regexes as manually specified features
  - **binary**: any matching sentence (for intents) + any word in a matching phrase (for slots)
    - **regexes meant to represent an intent/slot**
  - combination at different levels
    1) "input": aggregate word/sent + regex embeddings (at sentence level for intent, word level for slots)
    2) "network": per-label supervised attentions (log loss for regex matches)
    3) "output": alter final softmax (add weighted regex value)



- Good for limited amounts of data (few-shot)
  - works with 10-20 training examples per slot/intent
  - still improves a bit on full ATIS data

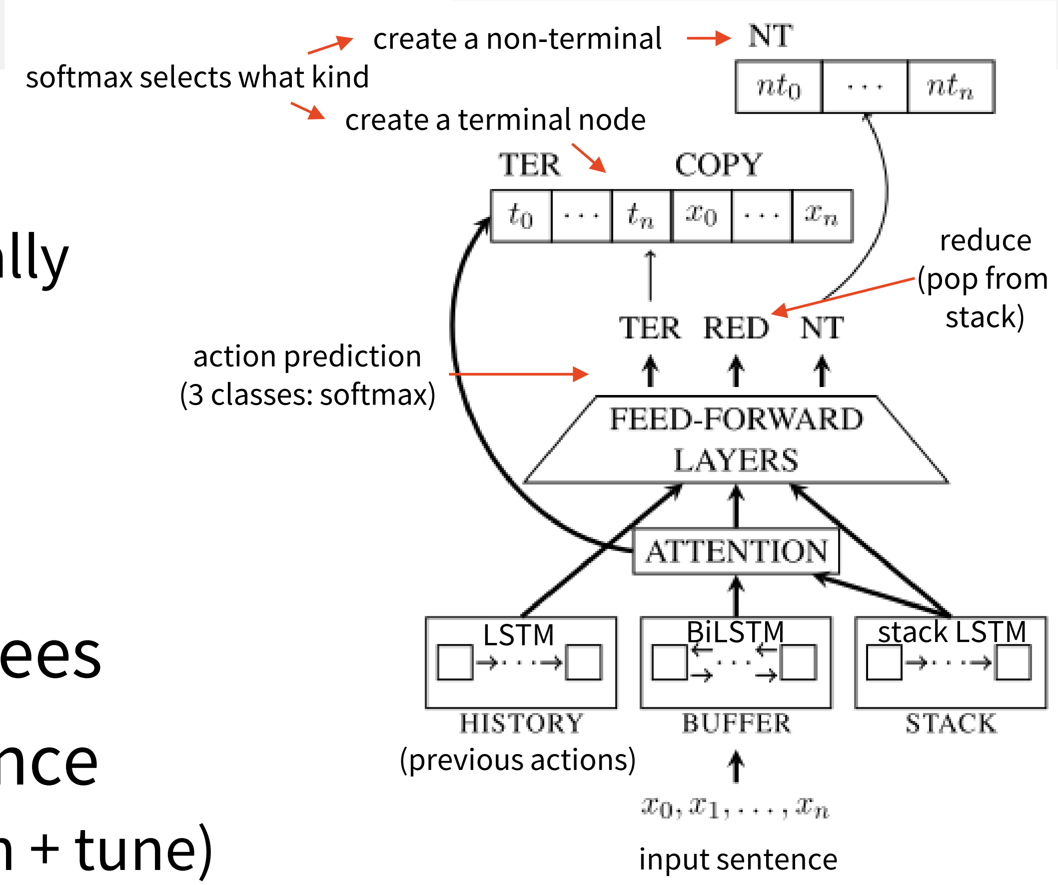
Model	Intent Macro-F1/Accuracy	Slot Macro-F1/Micro-F1
Liu&Lane (2016)	- / 98.43	- / 95.98
no regex (BiLSTM)	92.50 / 98.77	85.01 / 95.47
(1) input	91.86 / 97.65	86.7 / 95.55
(3) output	92.48 / 98.77	86.94 / 95.42
(2) network	**96.20 / 98.99**	85.44 / 95.27

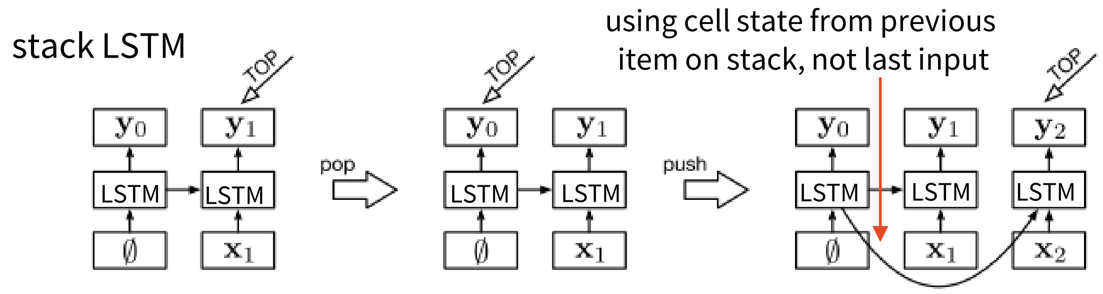


NLU as semantic parsing

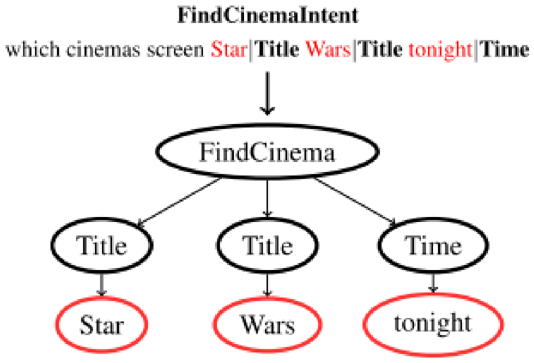
(Damonte et al., 2019) http://arxiv.org/abs/1903.04521

- **transition-based parsing**
  - actions over input build semantic tree gradually
  - using stack:
    - create terminal node (+ select what kind)
    - create non-terminal node (+ select what kind)
    - reduce – pop node from stack
- can parse into intent-slot-value shallow trees
- found to improve cross-domain performance
  - multi-task learning/transfer learning (pretrain + tune)





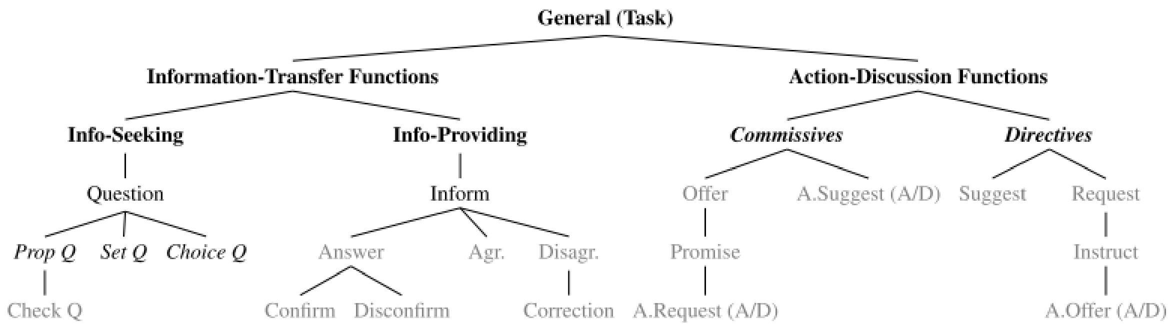
(Dyer et al, 2015)
http://arxiv.org/abs/1505.08075



Universal Intents

- typically DAs are domain-dependent

- **ISO** 24617-2 DA tagging **standard**
  - pretty complex: **multiple dimensions**
    - Task, Social, Feedback…
  - DA types (intents) under each dimension



(Mezza et al, 2018) https://www.aclweb.org/anthology/C18-1300

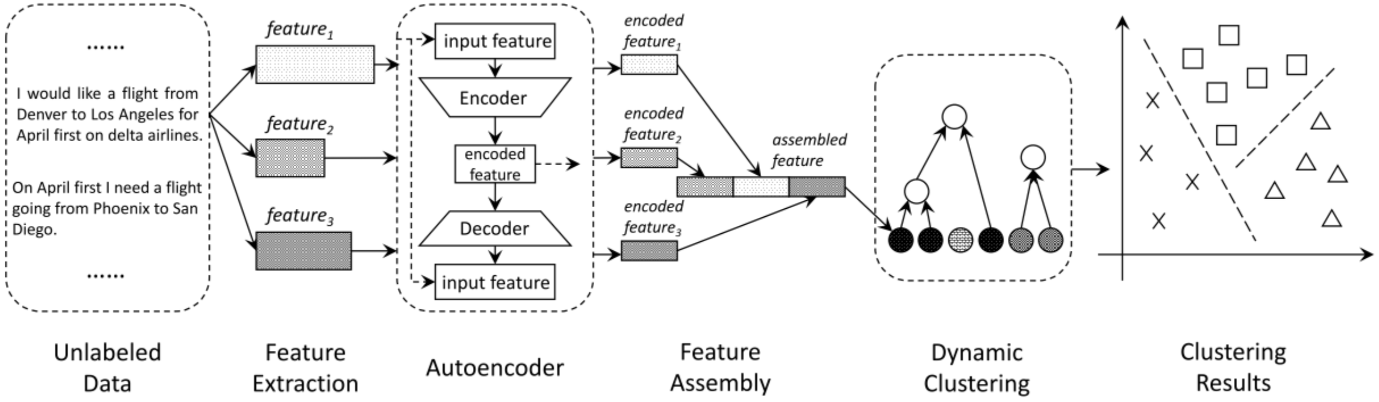
- **Simpler approach** – non-hierarchical
  - **union** looking at different datasets
  - Mapping from datasets – manual/semi-automatic
    - mapping tuned on classifier performance
  - Intent tagging improved using multiple datasets/domains
    - generic intents only
  - Slots stay domain-specific

ack, affirm, bye, deny, inform, repeat, reqalts, request, restart, thank-you, user-confirm, sys-impl-confirm, sys-expl-confirm, sys-hi, user-hi, sys-negate, user-negate, sys-notify-failure, sys-notify-success, sys-offer

(Paul et al, 2019)
http://arxiv.org/abs/1907.03020

Unsupervised NLU

- **Clustering** intents & slots
- Features:
  - word embeddings
  - POS
  - word classes
  - topic modelling (biterm)
- Autoencoder to normalize # of dimensions for features
- Dynamic hierarchical clustering
  - decides # of clusters – stops if cluster distance exceeds threshold
- Slot clustering – word-level
  - over nouns, using intent clustering results



feature choice + AE seem to work quite well

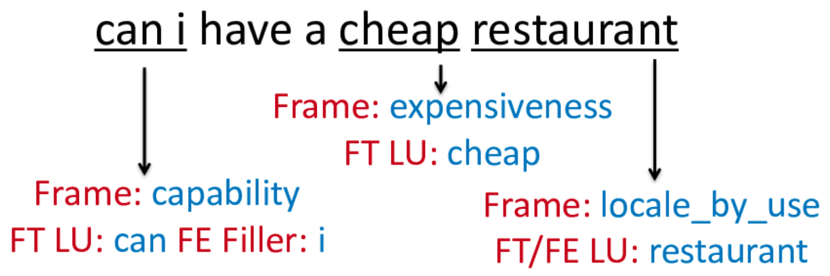
ATIS

Models	Intent Labeling Acc (%)
topic model	25.4
CDSSM vector	20.7
glove embedding	25.6
auto-dialabel	**84.1**

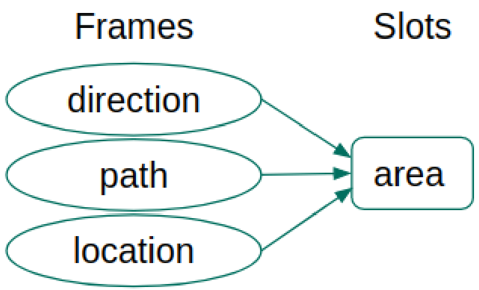
(Shi et al., 2018)
https://www.aclweb.org/anthology/D18-1072/

Unsupervised NLU with semantic frames (Vojta's work)

- Frame semantic parsing
  - Too general, not usable directly
  - Some frames *redundant*
  - Some frames *overlap*
- What about intents?

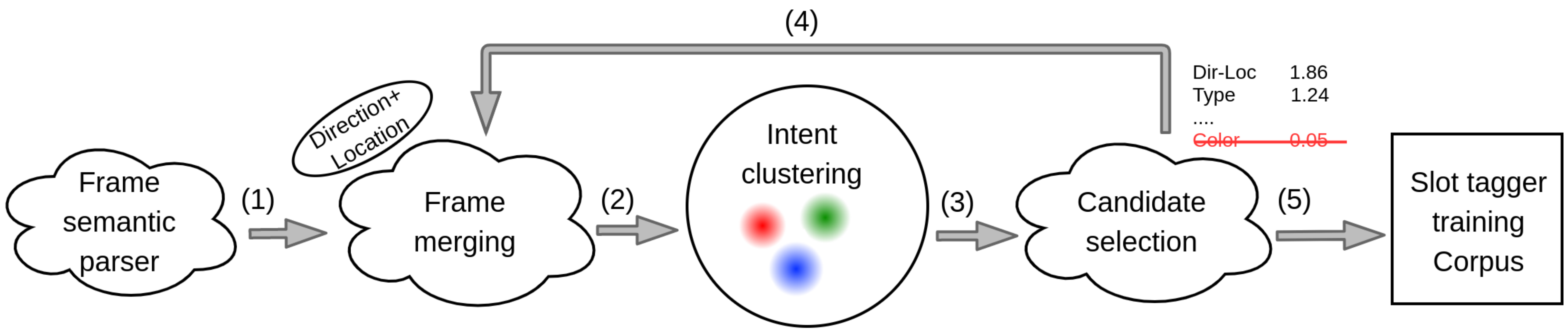


Redundancy



Overlap

Unsupervised NLU with semantic frames

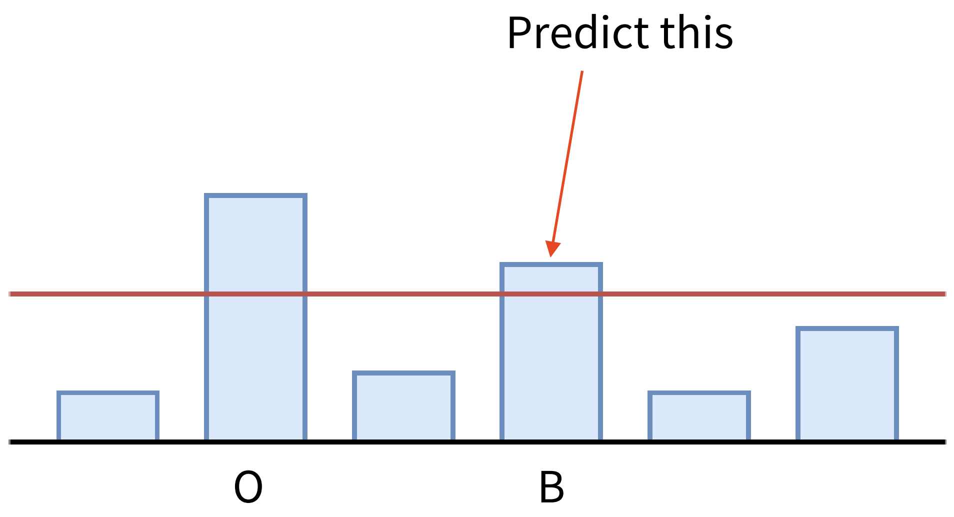


Unsupervised NLU with semantic frames - selection

- Iterative process
- Frames merging
  - Syntactic dependencies
  - 2 similar slots
- Candidates ranking
  - Based on frame semantic parser output
  - Multiple scoring functions (coherence, TextRank)

Unsupervised NLU with semantic frames - tagging

- LSTM B-I-O tagger
- Tagger trained on data previously labeled with our selection method
- Set threshold to improve recall



Unsupervised NLU with semantic frames - results

method	CamRest676	CarSLU	WOZ-hotel	WOZ-attraction	ATIS
Tag-supervised*	**0.778** ± .004	0.724 ± .003	**0.742** ± .008	**0.731** ± .002	**0.848** ± .003
Dict-supervised*	0.705 ± .005	**0.753** ± .005	**0.750** ± .018	0.665 ± .003	0.678 ± .002
Chen et al.	0.535 ± .002	0.590 ± .001	0.382 ± .001	0.375 ± .001	0.616 ± .001
Ours-nocl	0.311 ± .006	0.393 ± .011	0.122 ± .001	0.266 ± .008	0.631 ± .002
Ours-pars	0.552 ± .008	0.664 ± .007	0.388 ± .002	0.383 ± .002	0.627 ± .002
Ours-nothr	0.586 ± .024	0.569 ± .031	0.485 ± .032	0.435 ± .002	0.671 ± .005
Ours-full	**0.665** ± .012	**0.692** ± .008	**0.548** ± .004	**0.439** ± .001	**0.678** ± .002

Unsupervised NLU - drawbacks

- How to estimate the output quality?
- How to use the inducted slots?
  - What do they represent?
  - How to align with the DB?

Summary

- NLU is mostly **intent classification** + **slot tagging**
- **Rules + simple methods work well** with limited domains
- Neural NLU:
  - various architectures possible: CNN, LSTM, attention, seq2seq + pointer nets
  - slot tagging: **sequence prediction** – **label bias**
  - it helps to do joint intent + slots
  - BERT et al. can help, but these models are huge & expensive
    - there are specific pretrained dialogue models, too
  - NNs can be combined with regexes/handcrafted features
    - helps with limited data
- Experimental/alternative neural NLU:
  - using parsing (syntactic, semantic)
  - **unsupervised approaches**

Thanks

Contact us:

https://ufaldsg.slack.com/
{odusek,hudecek}@ufal.mff.cuni.cz
Skype/Meet/Zoom (by agreement)

Get the slides here:

http://ufal.cz/npfl099

Labs in 10 minutes
Dialmonkey Framework

Next Tue 9:50am:

- **State Tracking**
- **Lab Projects**

References/Inspiration/Further:

- mostly papers referenced from slides
- Milica Gašić's slides (Cambridge University): http://mi.eng.cam.ac.uk/~mg436/teaching.html
- Raymond Mooney's slides (University of Texas Austin): https://www.cs.utexas.edu/~mooney/ir-course/
- Filip Jurčíček's slides (Charles University): https://ufal.mff.cuni.cz/~jurcicek/NPFL099-SDS-2014LS/
- Hao Fang's slides (University of Washington): https://hao-fang.github.io/ee596_spr2018/syllabus.html
- Gokhan Tur & Renato De Mori (2011): Spoken Language Understanding