NPFL099 Statistical Dialogue Systems
5. Language Understanding

http://ufal.cz/npfl099

Ondřej Dušek, Zdeněk Kasner, Mateusz Lango, Ondřej Plátek

31. 10. 2024

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

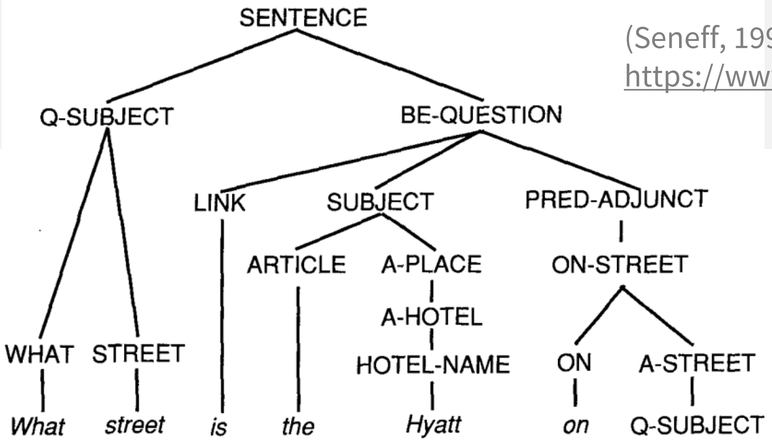
Natural Language Understanding

- **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)
- nowadays often just part of dialogue state tracking (next week)

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



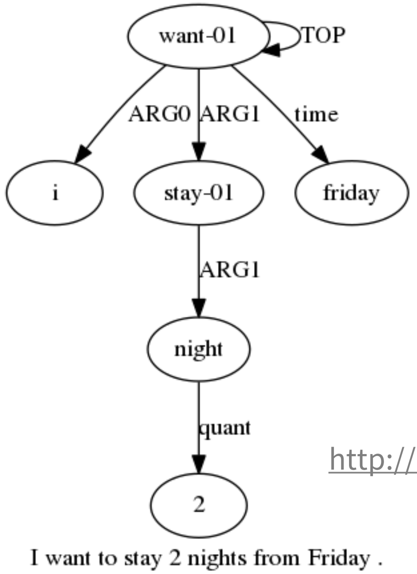
(Seneff, 1992)
https://www.aclweb.org/anthology/J92-1004

- 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

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

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

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**

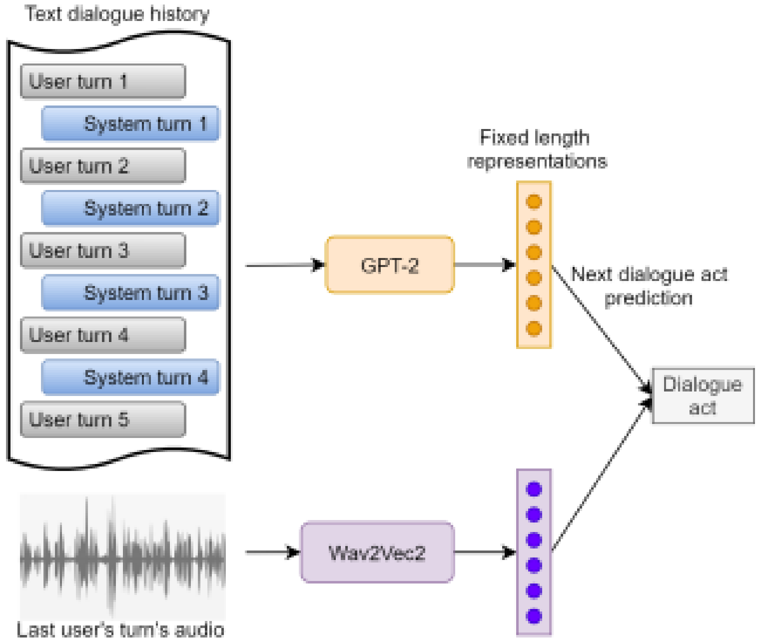
$$p(\text{DA}|\text{audio}) = \sum_{\text{texts}} P(\text{DA}|\text{text}) P(\text{text}|\text{audio})$$

- Alternative: **joint models**
  - in-domain ASR & NLU trained jointly
  - dual encoders, pretrained representations & combination

```
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()
```

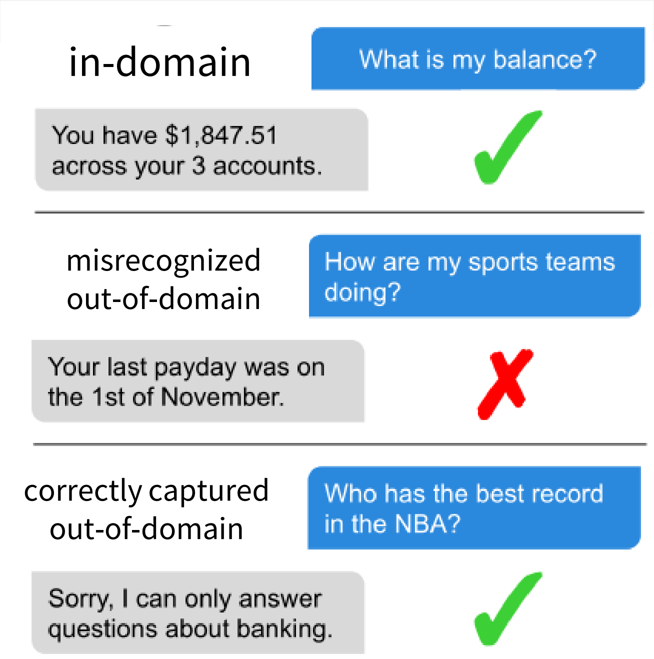
(from Filip Jurčíček's slides)



(Zorrilla et al., 2021) https://ieeexplore.ieee.org/document/9688296
(Si et al., 2023) http://arxiv.org/abs/2305.13040
(Rubenstein et al., 2023) http://arxiv.org/abs/2306.12925 `

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 -> ***<time>***
arrive_by -> ***none***

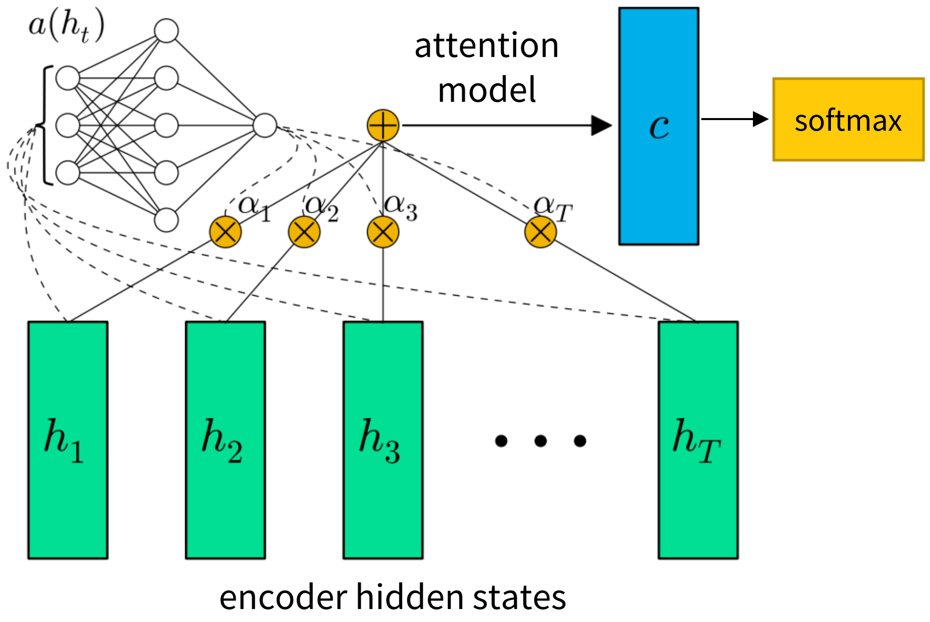
Both can be <time>

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

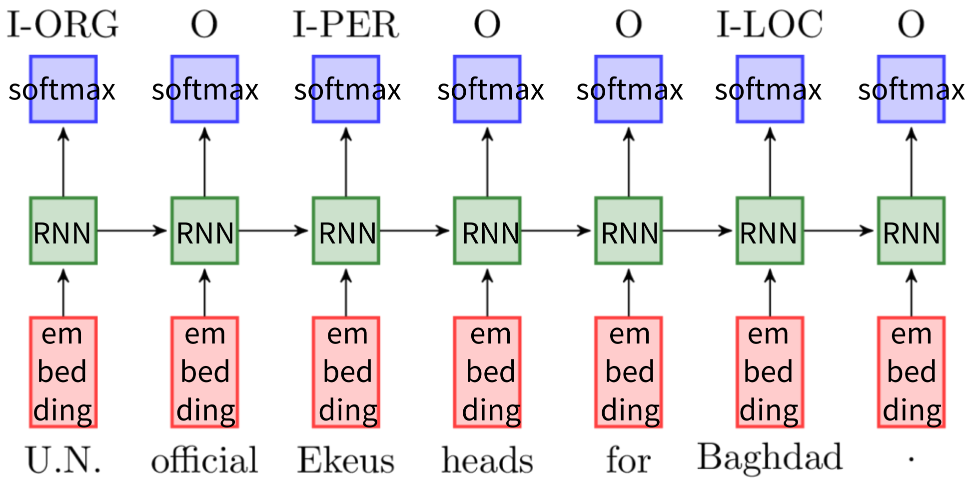
- get slot values directly – no need for delexicalization
  - each word classified
  - classes = slots & **IOB format** (inside-outside-beginning)
  - slot values taken from the text (where a slot is tagged)
  - NER-like approach

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

- 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

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

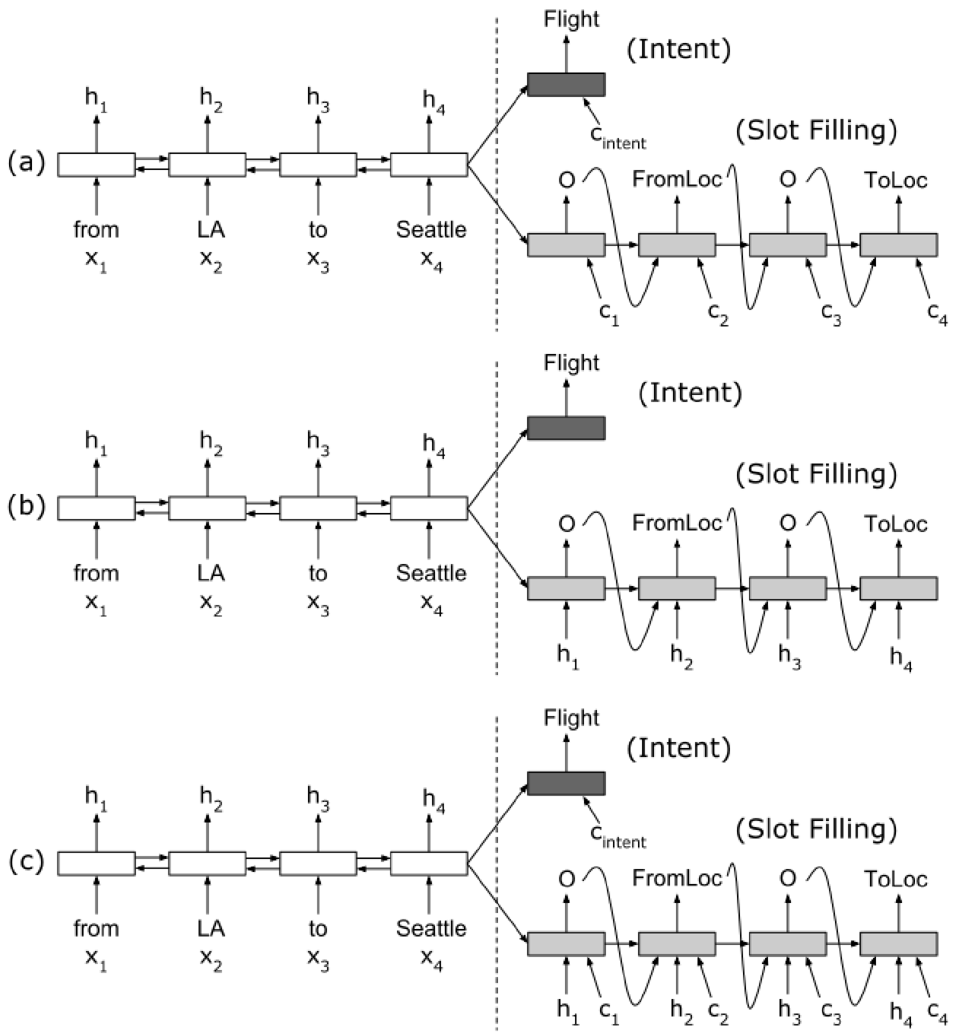


https://www.depends-on-the-definition.com/guide-sequence-tagging-neural-networks-python/

Joint Intent & Slots Model

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

- Same network for both tasks
- Bidirectional encoder
  - 2 RNN 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)

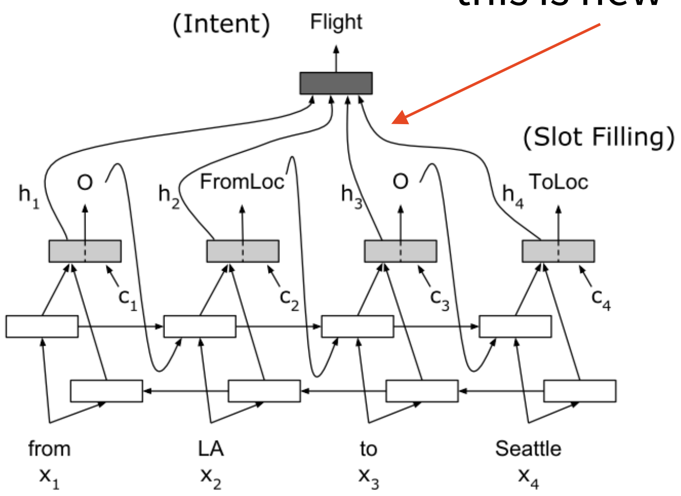


NN for Joint Intent & Slots

(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

this is new



same as (c)
on previous slide

- 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

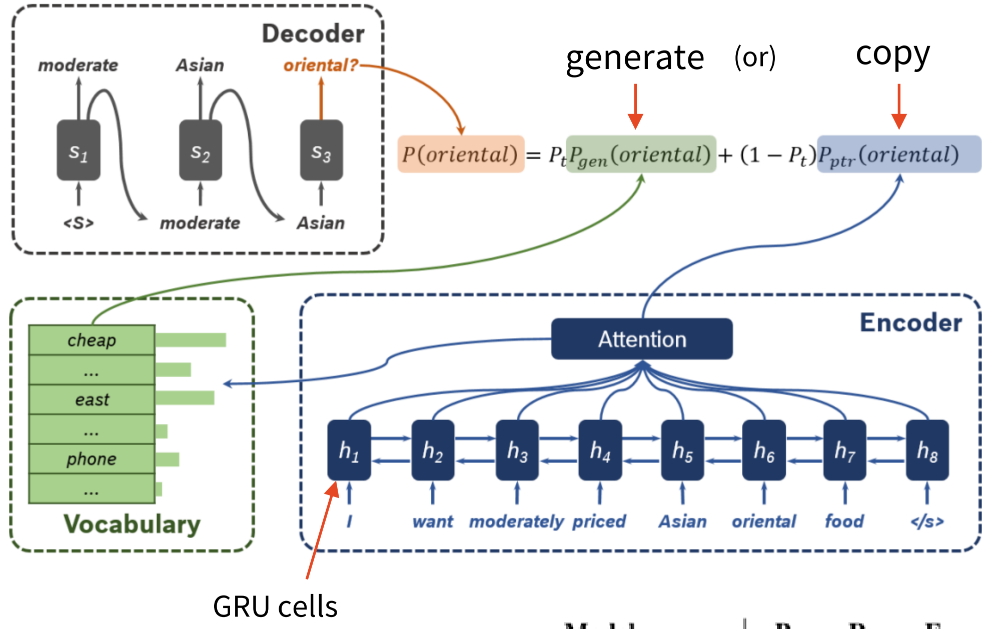
- Variant: treat **intent detection as slot tagging**
  - append <EOS> token & tag it with intent

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

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**

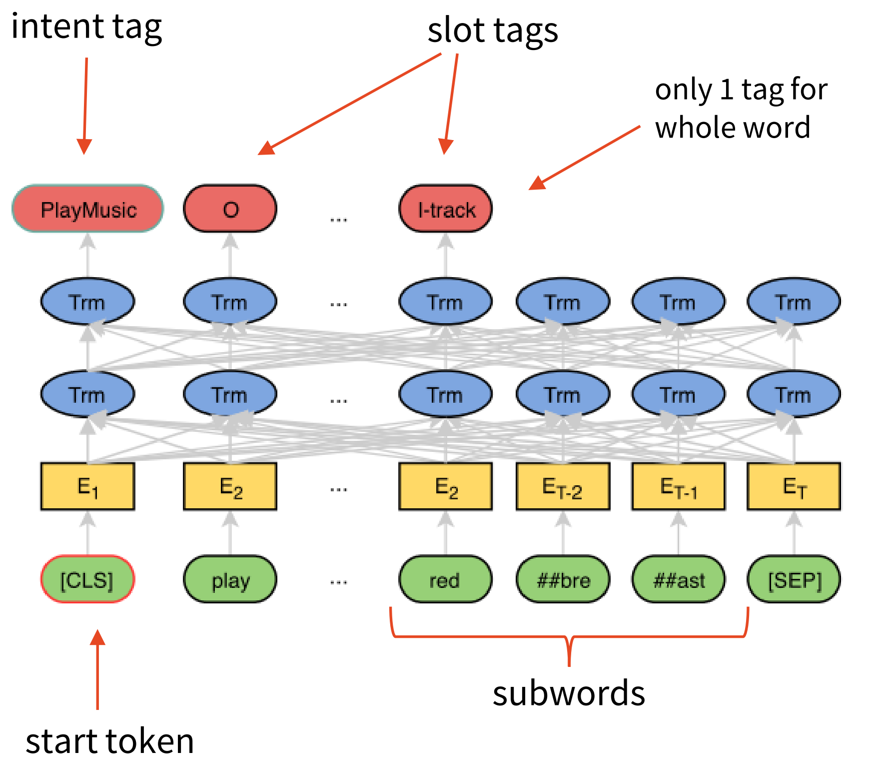
DSTC2 results

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			ATIS		
	Intent	Slot	Sent	Intent	Slot	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

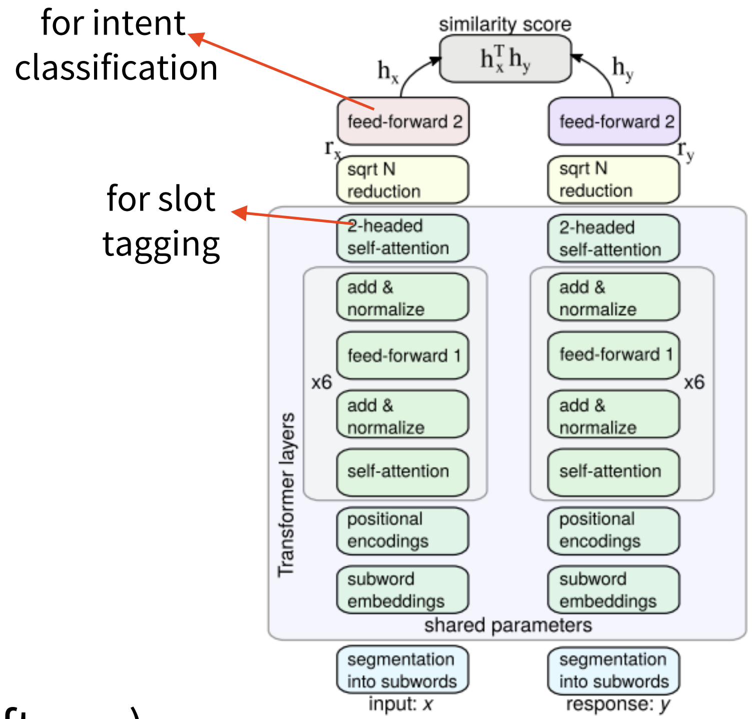
F1

Dialogue Pretrained Models

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

pre-LM | classif / seq tag

- 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**)

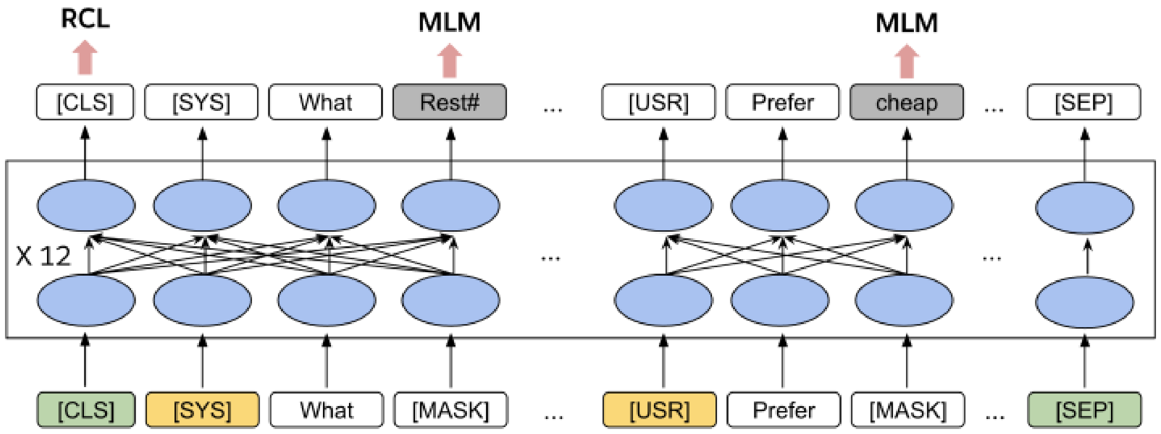


(Coope et al., 2020)
https://www.aclweb.org/anthology/2020.acl-main.11

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

TOD-BERT

- pre-finetuning BERT on vast *task-oriented* dialogue data
  - basically combination of 2 previous
- BERT + add user/sys tokens + train for:
  - masked language modelling
  - response selection (dual encoder style)
    - over [CLS] tokens from whole batch
    - other examples in batch = negative
- result: “better dialogue BERT”
  - can be finetuned for various dialogue tasks
    - intent classification
    - slot tagging
  - good performance even “few-shot”
    - just 1 or 10 examples per class
    - bigger difference w. r. t. BERT

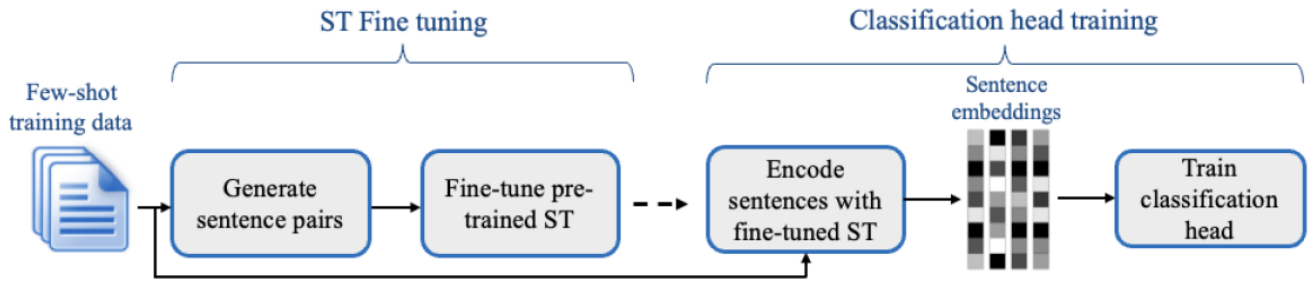


(Wu et al., 2020)
https://www.aclanthology.org/2020.emnlp-main.66

SETFIT: Sentence BERT + contrastive pre-finetuning

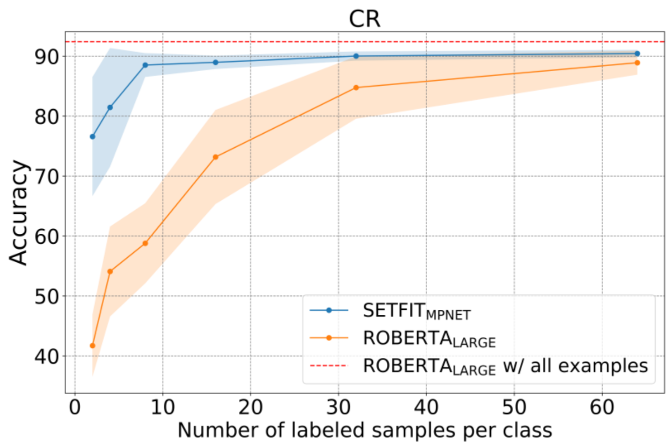
- Sentence Transformer (ST) = Transformer dual encoder
  - general, based on RoBERTa, produces sentence-level representations
  - trained for semantic similarity (NLI data)

(Reimers & Gurevych, 2019)
https://aclanthology.org/D19-1410/



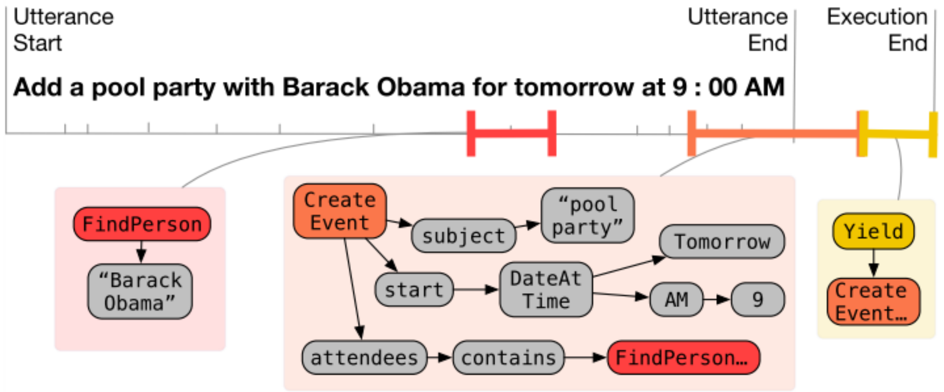
(Tunstall et al., 2022)
http://arxiv.org/abs/2209.11055

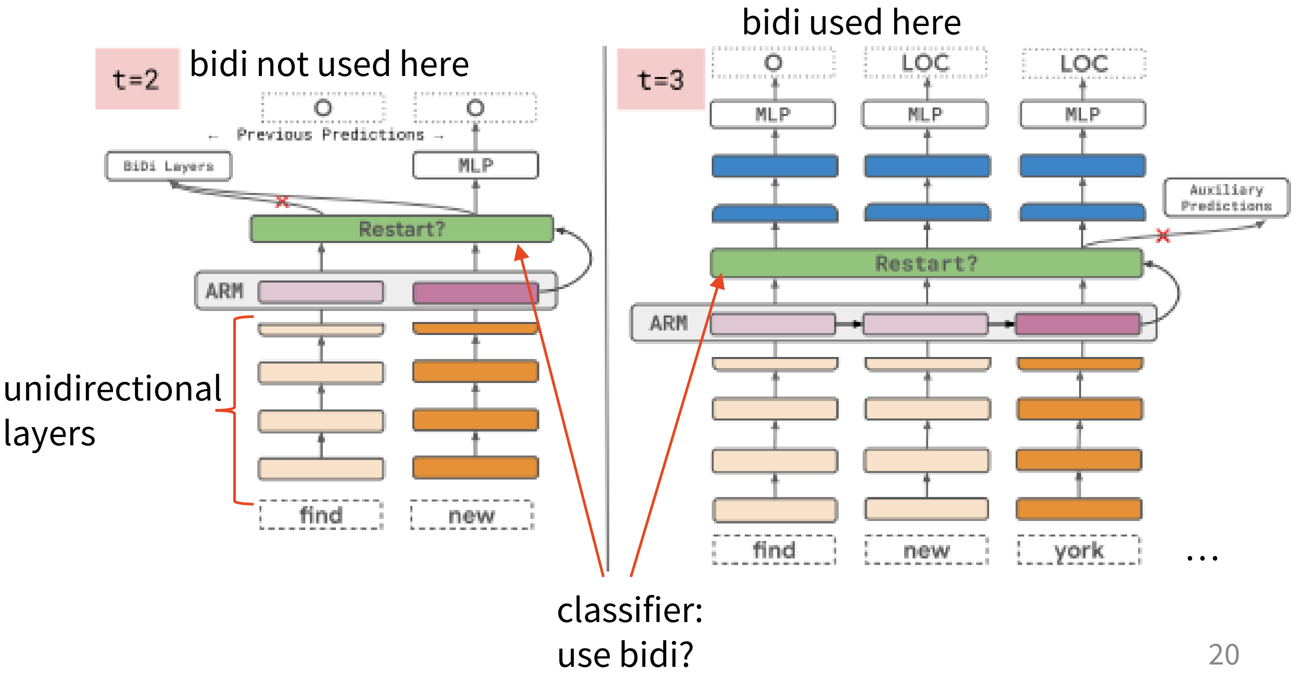
- Contrastive pre-finetuning:
  - 2 examples from same intent class = 1
  - 2 examples from random different intent classes = 0
- Intent classifier trained on top of the pre-finetuned model
- Good for low-data situations



Incremental NLU

- Aim: low latency, real-time performance
- Parsing incomplete sentences
  - guessing during parsing:
    create a full parse from incomplete sentences
  - predicting user input: use LM to finish utterance
  - both reduce latency
- Specific architecture
  - more like unidirectional encoders
    (so you don't need to recompute)
  - but retain bidirectional at higher layers
    - optionally, based on a specific classifier



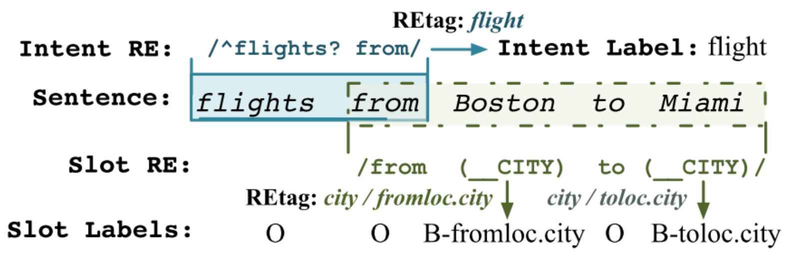


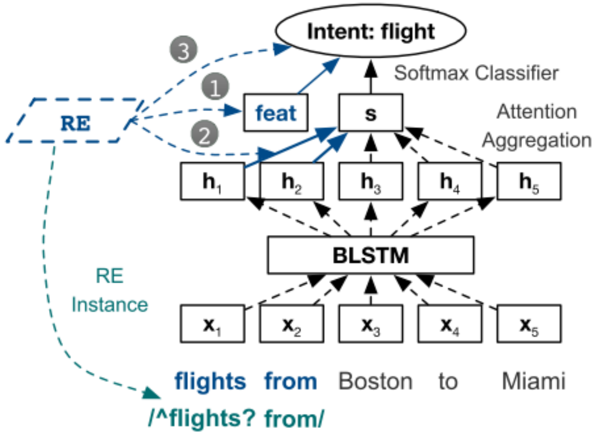
(Zhou et al., 2022) https://aclanthology.org/2022.acl-long.110
(Kaushal et al., 2023) https://aclanthology.org/2023.eacl-main.31

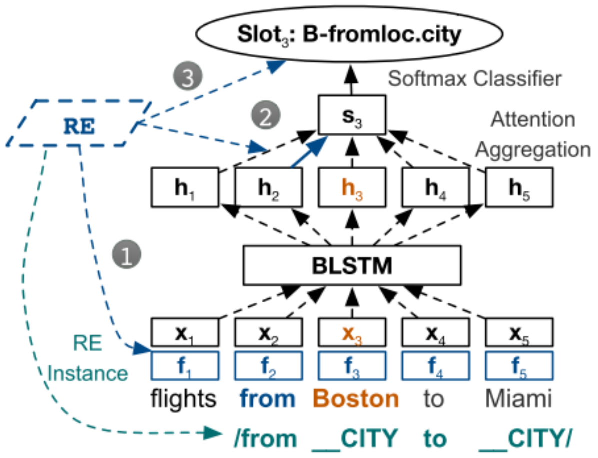
Regular Expressions & NNs for NLU

(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

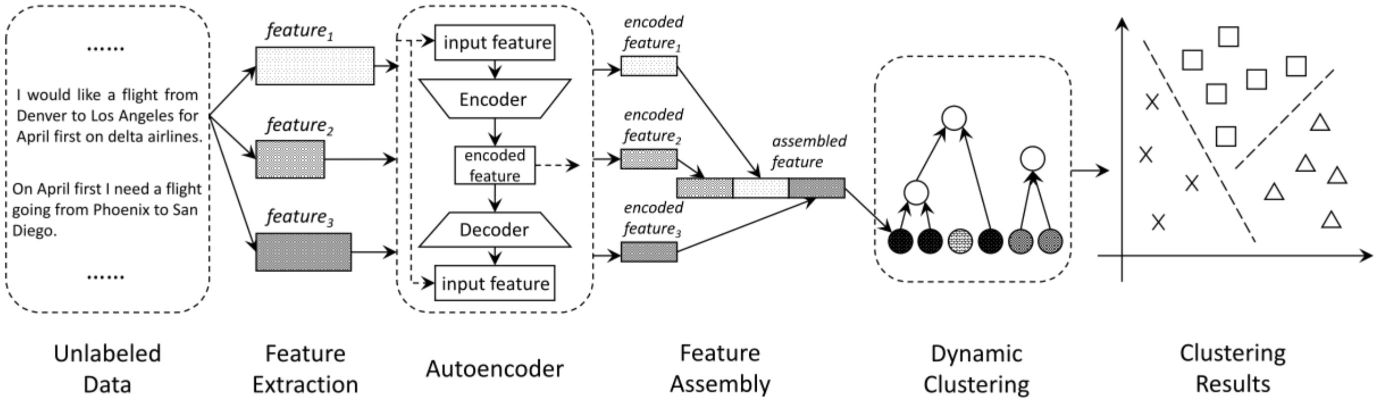






Unsupervised NLU

- **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

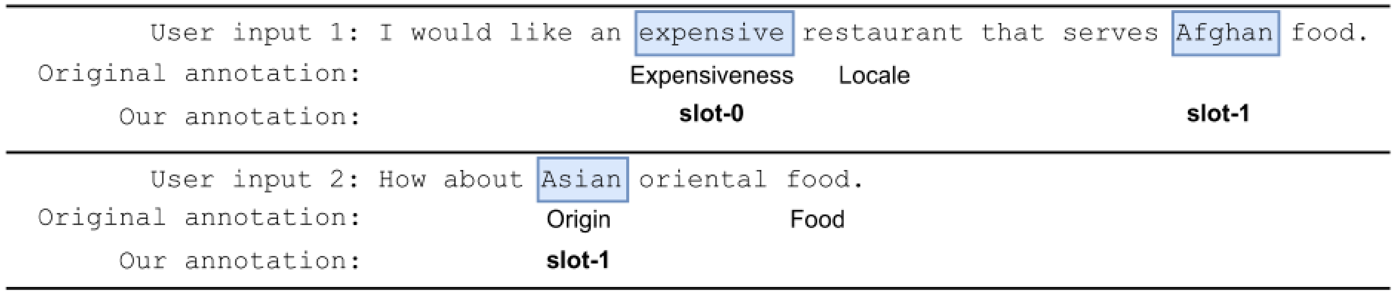
ATIS

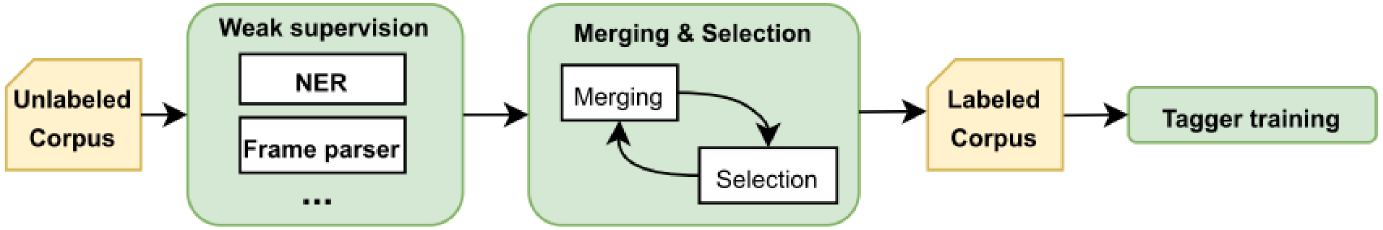
Models	Intent Labeling Acc (%)
topic model	25.4
CDSSM vector	20.7
glove embedding	25.6
auto-dialabel	**84.1**

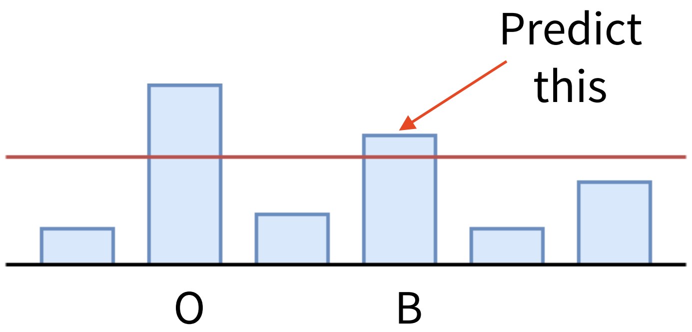
(Shi et al., 2018)
https://www.aclweb.org/anthology/D18-1072/

Weak Supervision from Semantic Frames

- Finding relevant **slots** based on **generic (frame) parser output**
  - filter irrelevant candidates, merge similar ones & generalize better
- Iterative merging & selection
  - frequency, coherence, TextRank
  - w. r. t. to head verbs
- Training an LSTM tagger
  - standalone, based on merged annotation
  - 2nd option threshold to improve recall
- Promising, but not perfect
  - DB connection, interpretation of slots







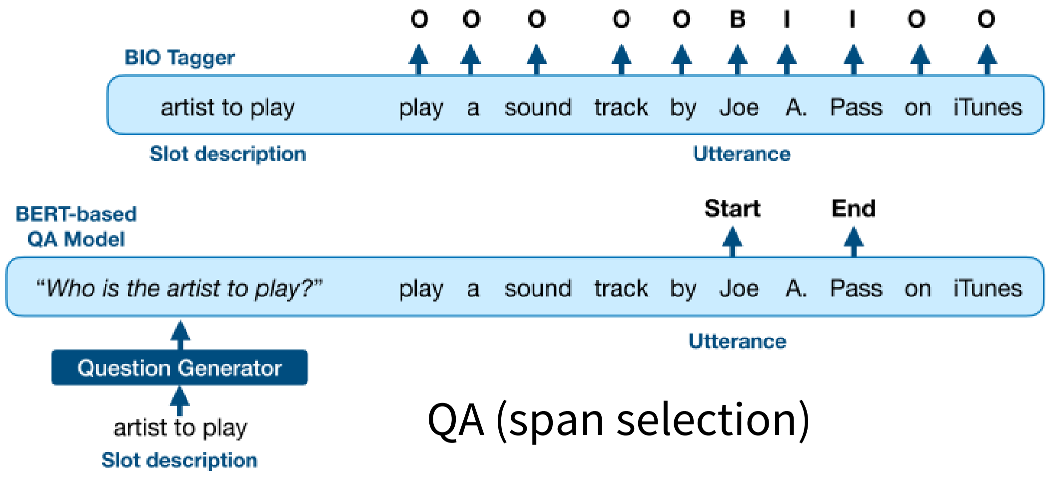
(Hudeček et al., 2021)
https://aclanthology.org/2021.acl-long.189

Weak supervision: QA-style NLU

- Zero-shot – just needs some slot descriptions
  - no in-domain training data needed
- Use a "question answering" BERT to do slot detection
  - generate questions from slot description – specifically ask for slots (rule-based)
  - QA model output = slot values
  - pretrained on other datasets (generate questions from ontology)
  - generalizes to unseen slots (though still far from perfect)

Slot	Raw Description	Our Question
playlist_owner	owner	who's the owner?
object_select	object select	which object to select?
best_rating	points in total	how many points in total?
num_book_people	number of people for booking	how many people for booking?

standard supervised slot tagger (for comparsion)



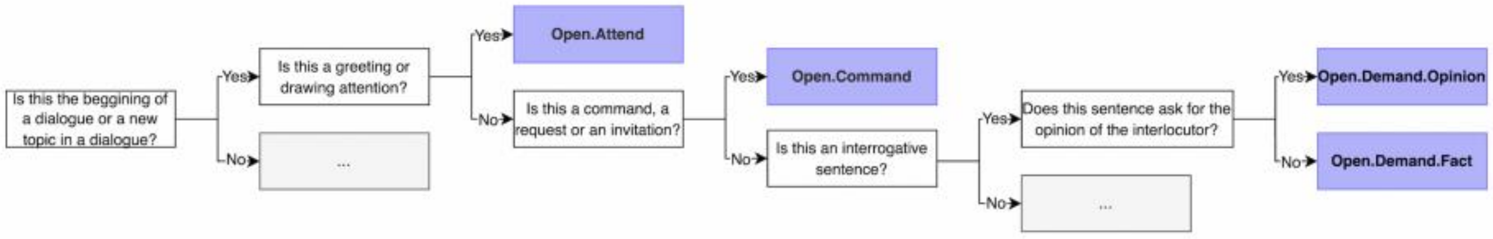
QA (span selection)

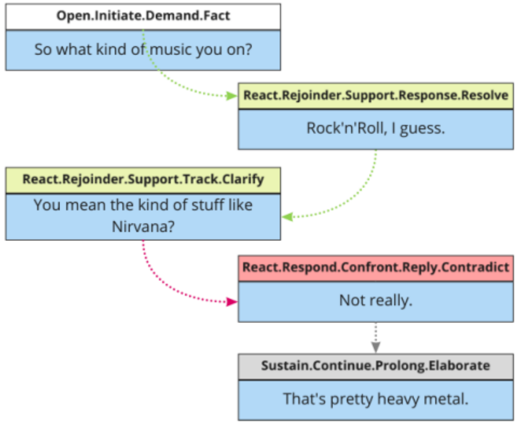
train: SNIPS, test: TOP	Zero-shot	Few-shot (20)	Few-shot (50)
Random NE	1.34	-	-
BERT seq tagging	8.82	37.60	42.73
BERT QA style	10.27	36.86	46.49
+ pretraining on other sets	12.35	39.78	47.91

(Du et al., 2021)
https://aclanthology.org/2021.acl-short.83

LLMs for (open-domain) NLU

- LLM prompts asking questions to:
  - classify sentences into a fixed schema
  - classify specific properties
- Prompt engineering
  - simple prompts
  - asking 1 question at a time
  - asking for reasoning
  - examples/not: depends

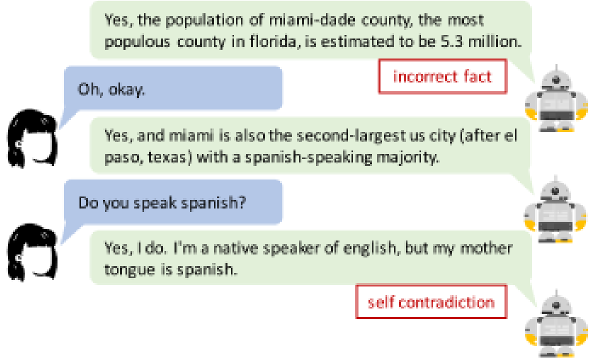




C	S1: Hi! S2: Hi! I am happy to be talking to you. Do you have any pets at home? S1: I don't, my mom is allergic to most pets. I used to have a pet fish when I was much younger. S2: Do you want a pet at some point in the future? No judgment here, I'm just curious. S1: I would love to have pets in the future! Once I have my own place, I'd like to get a dog or two.
Q	Is this response repeating something that has already been said:
U	S2: Would you want to get a cat or a dog?
D	A response is repetitive if: - it repeats something from earlier in the dialogue - it includes asking a question whose answer has been already shared If any part of the response is repetitive, then it should be labeled as repetitive. Note that sometimes repetition is useful, such as for emphasis, acknowledgement, clarification, or elaboration, and in these cases it should NOT be labeled as repetitive.

Provide your reasoning when considering this question starting with "Reasoning:". Then, finish by writing your final decision as one of: "Decision: [YES]" or "Decision: [NO]".

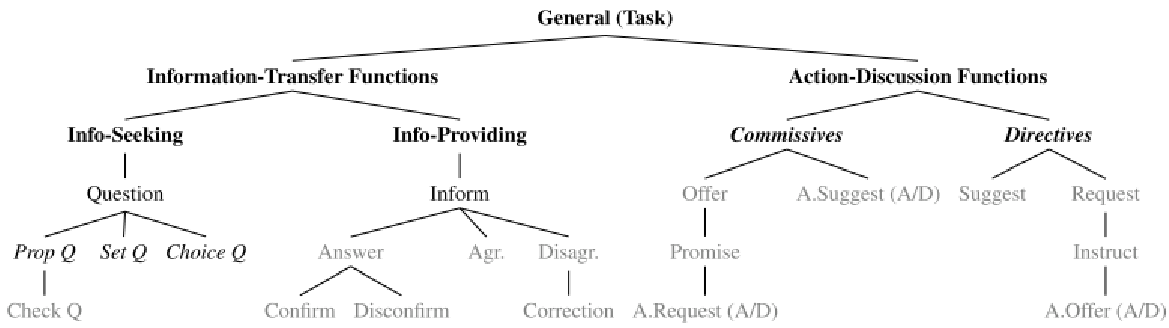
Empathetic
Lack of Empathy
Commonsense
Contradiction
Incorrect Fact
Self Contradiction
Partner Contradiction
Redundant
Ignore
Irrelevant



(Ostyakova et al., 2023) https://aclanthology.org/2023.sigdial-1.23
(Finch et al., 2023) https://aclanthology.org/2023.sigdial-1.20

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

Summary

- NLU is mostly **intent classification** + **slot tagging**
- **Rules + simple methods work well** with limited domains
- Neural NLU:
  - **shapes**: CNN, LSTM, attention, seq2seq + pointer nets
  - **tasks**: classification, sequence tagging, sequence prediction, span selection
  - it helps to do joint intent + slots
  - pretrained LMs help (models are large though)
    - BERT, specific pretrained dialogue models
  - NNs can be combined with regexes/handcrafted features
    - helps with limited data
- Less/no supervision: pretrained LMs & prompting, generic parsers, clustering
  - helps with domain generalization

Thanks

Contact us:

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

No labs today
Next week: lecture & labs

Get the slides here:

http://ufal.cz/npfl099

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