NPFL099 Statistical Dialogue Systems
5. Language Understanding

http://ufal.cz/npfl099

Ondřej Dušek, Vojtěch Hudeček & Zdeněk Kasner
31. 10. 2022
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

- disfluencies
  - hesitations – pauses, fillers, repetitions
  - fragments
  - self-repairs (~6%!)  

- ASR errors

- synonymy

- out-of-domain utterances

---

find something cheap for kids should be allowed

uhm I want something in the west the west part of town

uhm I’m looking for a cheap

uhm find something uhm something cheap no I mean moderate

I’m looking for a for a chip Chinese rest or rant

Chinese city centre

I’ve been wondering if you could find me a restaurant that has Chinese food close to the city centre please

oh yeah I’ve heard about that place my son was there last month
Semantic representations

- **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")`

---

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

*Oui l'hôtel don't le prix ne dépasse pas cent dix euros*  
(Bonneau-Maynard et al., 2005)  
[https://www.isca-speech.org/archive/interspeech_2005/i05_3457.html](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**

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

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

![Graph showing probabilities and word skipping](from Filip Jurčíček's slides)
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)
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)
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 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

• 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 $c_{\text{intent}}$ (attention)
NN for Joint Intent & Slots

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

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

(Hakkani-Tür et al, 2016)
https://doi.org/10.21437/Interspeech.2016-402
Seq2seq-based NLU

• 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

---

Can I bring my kids along to this restaurant?
I want a Chinese place with a takeaway option.

confirm(kids_friendly=yes)
inform(food=Chinese_takeaway)

---

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
BERT-based NLU

- 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

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

(NPFL099 L5 2022)

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

Accuracy

F1

% completely correct sentences

start token

subwords

intent tag

slot tags

only 1 tag for whole word

pre-LM | seq tag
Dialogue Pretrained Models

- 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**)

---

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

(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
• 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)

- 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

---

See also:

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

- (Tunstall et al., 2022)
Regular Expressions & NNs for NLU

- 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

---

<table>
<thead>
<tr>
<th>Model</th>
<th>Intent Micro-F1/Accuracy</th>
<th>Slot Micro-F1/Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu&amp;Lane (2016)</td>
<td>- / 98.43</td>
<td>- / 95.98</td>
</tr>
<tr>
<td>no regex (BiLSTM)</td>
<td>92.50 / 98.77</td>
<td>85.01 / 95.47</td>
</tr>
<tr>
<td>(1) input</td>
<td>91.86 / 97.65</td>
<td>86.7 / 95.85</td>
</tr>
<tr>
<td>(3) output</td>
<td>92.48 / 98.77</td>
<td>86.94 / 95.42</td>
</tr>
<tr>
<td>(2) network</td>
<td><strong>96.20 / 98.99</strong></td>
<td>85.44 / 95.27</td>
</tr>
</tbody>
</table>

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

(Shi et al., 2018)
https://www.aclweb.org/anthology/D18-1072/
Weak Supervision from Semantic Frames
(Vojta’s work)

• 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
  • 2\textsuperscript{nd} 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)

<table>
<thead>
<tr>
<th>Slot</th>
<th>Raw Description</th>
<th>Our Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>playlist_owner</td>
<td>owner</td>
<td>who’s the owner?</td>
</tr>
<tr>
<td>object_select</td>
<td>object select</td>
<td>which object to select?</td>
</tr>
<tr>
<td>best_rating</td>
<td>points in total</td>
<td>how many points in total?</td>
</tr>
<tr>
<td>num_book_people</td>
<td>number of people for booking</td>
<td>how many people for booking?</td>
</tr>
</tbody>
</table>

(Bio Tagger)

(BERT QA Model)

(Question Generator)

(train: SNIPS, test: TOP)

<table>
<thead>
<tr>
<th></th>
<th>Zero-shot</th>
<th>Few-shot (20)</th>
<th>Few-shot (50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random NE</td>
<td>1.34</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BERT seq tagging</td>
<td>8.82</td>
<td>37.60</td>
<td>42.73</td>
</tr>
<tr>
<td>BERT QA style</td>
<td>10.27</td>
<td>36.86</td>
<td>46.49</td>
</tr>
<tr>
<td>+ pretraining on other sets</td>
<td>12.35</td>
<td>39.78</td>
<td>47.91</td>
</tr>
</tbody>
</table>

standard supervised slot tagger (for comparison)

QA (span selection)

(Du et al., 2021)

https://aclanthology.org/2021.acl-short.83
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

- **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

(Mezza et al, 2018) [https://www.aclweb.org/anthology/C18-1300](https://www.aclweb.org/anthology/C18-1300)
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, generic parsers, clustering
  • helps with domain generalization
Thanks

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

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/
• 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

No labs today
Next week: lecture & labs