NPFL099 Statistical Dialogue Systems
5. Language Understanding

http://ufal.cz/npfl099

Ondřej Dušek, Vojtěch Hudeček & Tomáš Nekvinda
1. 11. 2021
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

- **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

\[
\text{inform(date=Friday, stay="2 nights")}
\]
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: **confusion nets** with weighted words
  - a more concise way of showing the same thing

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)

(left-to-right: multiply probabilities)
Handling out-of-domain queries

- Handcrafted: **no pattern matches** \(\rightarrow\) out-of-domain
- Datasets – rarely taken into account!
- **Low confidence** on any intent \(\rightarrow\) 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
```

```
0 0 0 0 B-dept 0 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 $c_{intent}$ (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
Joint intents & slots with contextual embeddings

• 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: $P(l_i|h_i)$
    • label-recurrent – depend on past labels & current embedding: $P(l_i|l_{1,...,i-1}, h_i)$
      • faster than word-recurrent

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

<table>
<thead>
<tr>
<th>Model</th>
<th>label recurrent</th>
<th>intent classif. accuracy</th>
<th>slot labelling F1</th>
<th>Inference time (ms/utterance)</th>
<th>Epochs to converge</th>
<th>s/epoch</th>
<th># params</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEED-FORWARD</td>
<td>No</td>
<td>98.56 97.14</td>
<td>53.39 69.68</td>
<td>0.61</td>
<td>48</td>
<td>1.82</td>
<td>17k</td>
</tr>
<tr>
<td>FEED-FORWARD</td>
<td>Yes</td>
<td>98.54 97.46</td>
<td>75.35 88.72</td>
<td>1.82</td>
<td>83</td>
<td>2.52</td>
<td>19k</td>
</tr>
<tr>
<td>CNN, 5KERNEL, 1L</td>
<td>No</td>
<td>98.56 98.40</td>
<td>85.88 94.11</td>
<td>0.82</td>
<td>23</td>
<td>1.90</td>
<td>42k</td>
</tr>
<tr>
<td>CNN, 5KERNEL, 3L</td>
<td>No</td>
<td>99.04 98.42</td>
<td>92.21 96.68</td>
<td>1.37</td>
<td>55</td>
<td>2.16</td>
<td>91k</td>
</tr>
<tr>
<td>CNN, 5KERNEL, 4L</td>
<td>No</td>
<td>98.81 98.32</td>
<td>91.65 96.75</td>
<td>1.28</td>
<td>57</td>
<td>2.97</td>
<td>76k</td>
</tr>
<tr>
<td>CNN, 5KERNEL, 1L</td>
<td>Yes</td>
<td>98.85 98.36</td>
<td>93.12 96.39</td>
<td>2.13</td>
<td>51</td>
<td>2.77</td>
<td>43k</td>
</tr>
<tr>
<td>CNN, 5KERNEL, 3L</td>
<td>Yes</td>
<td>99.10 98.36</td>
<td>94.22 96.95</td>
<td>2.68</td>
<td>59</td>
<td>3.34</td>
<td>93k</td>
</tr>
<tr>
<td>CNN, 5KERNEL, 4L</td>
<td>Yes</td>
<td>98.90 98.32</td>
<td>93.71 96.95</td>
<td>2.60</td>
<td>53</td>
<td>3.43</td>
<td>78k</td>
</tr>
<tr>
<td>ATTN, 1HEAD, 1L</td>
<td>No</td>
<td>98.50 97.51</td>
<td>83.61 69.31</td>
<td>1.95</td>
<td>25</td>
<td>1.94</td>
<td>22k</td>
</tr>
<tr>
<td>ATTN, 1HEAD, 1L</td>
<td>No</td>
<td>98.53 97.74</td>
<td>75.55 93.22</td>
<td>4.75</td>
<td>117</td>
<td>4.34</td>
<td>23k</td>
</tr>
<tr>
<td>ATTN, 1HEAD, 3L</td>
<td>No</td>
<td>98.74 98.10</td>
<td>81.51 94.07</td>
<td>7.68</td>
<td>160</td>
<td>4.32</td>
<td>33k</td>
</tr>
<tr>
<td>ATTN, 2HEAD, 3L</td>
<td>No</td>
<td>98.31 98.10</td>
<td>83.02 94.61</td>
<td>7.86</td>
<td>79</td>
<td>4.87</td>
<td>47k</td>
</tr>
<tr>
<td>ATTN, 1HEAD, 1L</td>
<td>Yes</td>
<td>98.63 97.68</td>
<td>74.94 88.60</td>
<td>3.24</td>
<td>60</td>
<td>2.66</td>
<td>21k</td>
</tr>
<tr>
<td>ATTN, 1HEAD, 1L</td>
<td>Yes</td>
<td>98.61 98.00</td>
<td>86.72 94.53</td>
<td>6.12</td>
<td>89</td>
<td>5.53</td>
<td>24k</td>
</tr>
<tr>
<td>ATTN, 1HEAD, 3L</td>
<td>Yes</td>
<td>98.51 98.26</td>
<td>88.04 94.99</td>
<td>9.03</td>
<td>109</td>
<td>6.06</td>
<td>34k</td>
</tr>
<tr>
<td>ATTN, 2HEAD, 3L</td>
<td>Yes</td>
<td>98.48 98.26</td>
<td>89.31 95.86</td>
<td>9.17</td>
<td>93</td>
<td>6.54</td>
<td>49k</td>
</tr>
<tr>
<td>LSTM, 1L</td>
<td>No</td>
<td>98.82 98.34</td>
<td>91.83 97.28</td>
<td>2.65</td>
<td>45</td>
<td>2.91</td>
<td>47k</td>
</tr>
<tr>
<td>LSTM, 2L</td>
<td>No</td>
<td>98.77 98.30</td>
<td>93.10 97.36</td>
<td>4.72</td>
<td>57</td>
<td>5.09</td>
<td>77k</td>
</tr>
<tr>
<td>LSTM, 1L</td>
<td>Yes</td>
<td>98.68 98.36</td>
<td>93.83 97.27</td>
<td>3.68</td>
<td>54</td>
<td>4.62</td>
<td>49k</td>
</tr>
<tr>
<td>LSTM, 2L</td>
<td>Yes</td>
<td>98.71 98.30</td>
<td>93.88 97.28</td>
<td>6.03</td>
<td>69</td>
<td>6.82</td>
<td>79k</td>
</tr>
</tbody>
</table>

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

*NN | classif + seq tag
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)**

---

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

<table>
<thead>
<tr>
<th>Models</th>
<th>Intent</th>
<th>Snips Slot</th>
<th>Sent</th>
<th>Intent</th>
<th>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>98.6</td>
<td>97.0</td>
<td>92.8</td>
<td>97.5</td>
<td>96.1</td>
<td>88.2</td>
</tr>
<tr>
<td>Joint BERT + CRF</td>
<td>98.4</td>
<td>96.7</td>
<td>92.6</td>
<td>97.9</td>
<td>96.0</td>
<td>88.6</td>
</tr>
</tbody>
</table>

Slightly different numbers, most probably a reimplementation

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

Accuracy F1

% completely correct sentences

Start token

Intent tag

Slot tags

Only 1 tag for whole word

Subwords
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-LM classif / seq tag
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
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

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

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

standard supervised slot tagger (for comparison)

QA (span selection)

<table>
<thead>
<tr>
<th>Slot description</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>artist to play</td>
<td>play a sound track by Joe A. Pass on iTunes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BIO Tagger</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 B I I I O O</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>“Who is the artist to play?”</th>
<th>play a sound track by Joe A. Pass on iTunes</th>
</tr>
</thead>
</table>

(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

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}@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

Labs in 10 minutes
Dialmonkey Framework

Next Tue 9:50am:
• State Tracking
• Lab Projects