Traitement Automatique du Langage Naturel pour la Fouille de Texte
Natural Language Understanding

Marco Dinarelli
(exploiting slides of François Portet)

M2 MOSIG

December 12, 2022

Sources

Neumann and Xu, **Essli Summer School**

Dan Jurafsky and James H. Martin, **Speech and Language Processing (3rd ed. draft)** [Jurafsky and Manning, 2021]

Natural Language Understanding

NLU as sequence labelling task

Named Entity Recognition (NER)

Slot Filling

Natural Language Understanding (NLU)

Old sub-field of NLP devoted to extract *semantics* from texts (STUDENT [Bobrow, 1964], SHRDLU[Winograd, 1970])

Natural Language Understanding (NLU)

Old sub-field of NLP devoted to extract *semantics* from texts (STUDENT [Bobrow, 1964], SHRDLU[Winograd, 1970])

Usual tasks

- Text classification (e.g., junk/not junk)
- Question answering (e.g., ‘when is the lock-down going to end?’)
- Named Entity Recognition (e.g., ‘New York’ is a place, ‘Mr Smith’ is a person ...)
- Entity linking (e.g., ‘New York’ is related to the NY ressource...)
- Relation Extraction (e.g., ‘the parcel is behind the wall’ → behind_of(parcel_1,wall_2))
- Topic recognition (e.g., ‘Paris won’ → sport, ‘new antibiotic’ → biology ...)
- Sentiment analysis (e.g., ‘awful room’ → negative ...)
- ...

Natural Language Understanding (NLU)

Old sub-field of NLP devoted to extract *semantics* from texts (STUDENT [Bobrow, 1964], SHRDLU[Winograd, 1970])

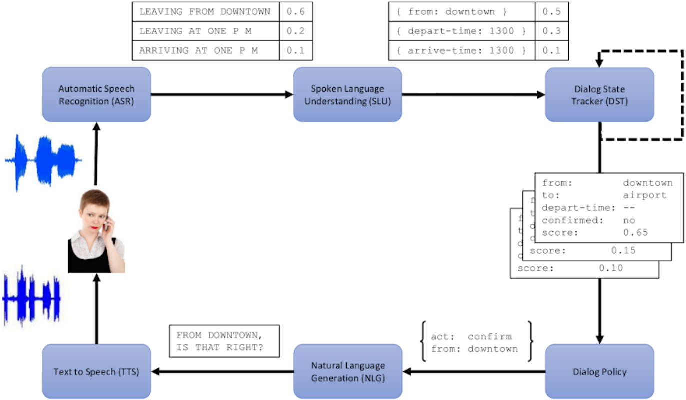
Usual tasks

- Text classification (e.g., junk/not junk)
- Question answering (e.g., 'when is the lock-down going to end?')
- Named Entity Recognition (e.g., 'New York' is a place, 'Mr Smith' is a person ...)
- Entity linking (e.g., 'New York' is related to the NY ressource...)
- Relation Extraction (e.g., 'the parcel is behind the wall' → behind_of(parcel_1,wall_2))
- Topic recognition (e.g., 'Paris won' → sport, 'new antibiotic' → biology ...)
- Sentiment analysis (e.g., 'awful room' → negative ...)
- ...

Main challenges

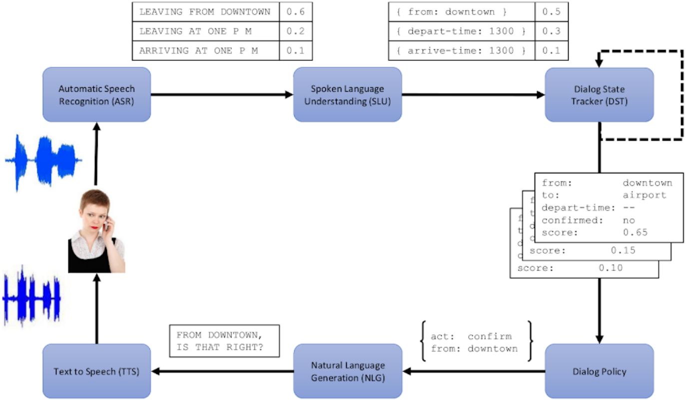
- Ambiguity
- Context handling
- Open vs Close World

NLU and dialogue



after [Williams et al., 2016]

NLU and dialogue



after [Williams et al., 2016]

Most systems are:

- Very restricted in term of task and semantic space
- Dyadic
- Pipeline

Examples

- Slot filling: LUIS
- Entity Linking: TagME

Natural Language Understanding

NLU as sequence labelling task

Named Entity Recognition (NER)

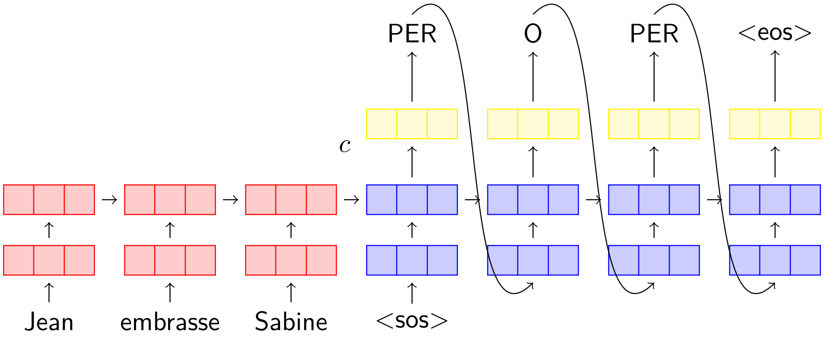
Slot Filling

Example with ATIS [Hemphill et al., 1990]

Sentence	*show*	*flights*	*from*	*Boston*	*To*	*New*	*York*	*today*
Slots/Concepts	O	O	O	B-dept	O	B-arr	I-arr	B-date
Named Entity	O	O	O	B-city	O	B-city	I-city	O
Intent	*Find_Flight*							
Domain	*Airline Travel*							

- Intent recognition influenced by speech act theory [Searle and Searle, 1969]
  → often addressed as a classification problem
- Slot/Concept influenced by Frame semantics [Fillmore et al., 1976]
  → often addressed as a sequence labelling problem

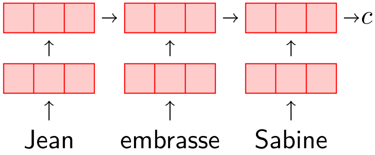
Encoder decoder architecture [Sutskever et al., 2014]



The input is summarized by one single vector by the encoder

The output is generated from this single vector by the decoder

Encoder



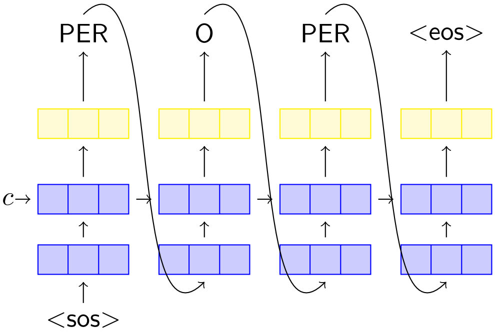
an encoder reads the input sequence of vectors $\mathbf{x} = (x_1, \cdots, x_T)$ and generates hidden states $h_t = f(x_t, h_{t-1}) \in \mathbf{R}^n$
c is a context vector generated from the sequence of hidden states

$$c = q(\{h_1, \cdots, h_T\}),$$

f and q are some nonlinear functions.

Most often f is a RNN (LSTM, GRU) and q is chosen such that $c = q(\{h_1, \cdots, h_T\}) = h_T$ [Sutskever et al., 2014].

decoder



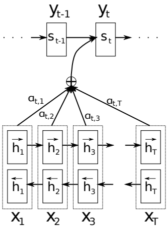
The decoder predicts the next word y_t given c and $\{y_1, \cdots, y_{t-1}\}$.

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

where g is a non-linear function that outputs the probability of y_t, and s_t is the hidden state of the RNN.

Attention mechanism

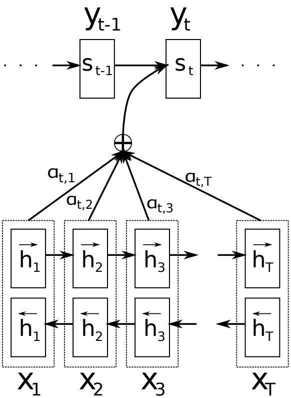
[Bahdanau et al., 2015]



$$p(y_t|y_1,\ldots,y_{t-1},\mathbf{x}) = g(y_{t-1}, s_t, c_t), \text{ where } s_t = f(s_{t-1}, y_{t-1}, c_t).$$

problem: the context c can be too abstracted for the task. Some information from the input can be lost.
idea: use a weighted sum of the input in the context vector c
use the input hidden state. Indeed, h_i contains information about the whole input sequence with a strong focus on the parts surrounding the t-th word of the input sequence.

Attention mechanism



$$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j.$$

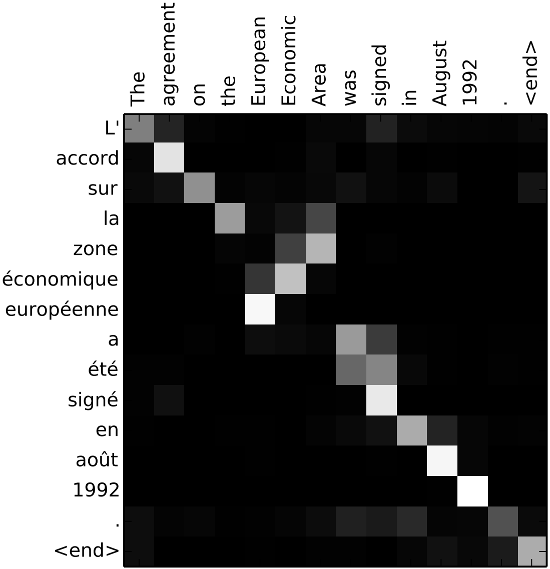
The weight α_{tj} is computed by

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T} \exp(e_{tk})},$$

where $e_{tj} = a(s_{t-1}, h_j)$

$a()$ is an *alignment model* which scores how well the inputs around position j and the output at position i match. a is trained by feedforward neural network which is jointly trained with all the other components of the model.

Attention – Alignment



Natural Language Understanding

NLU as sequence labelling task

Named Entity Recognition (NER)

Slot Filling

Named Entity Recognition

Task

Identify text segments expressing references to named entities (NE) including

- person names
- company/organization names
- locations

Named Entity Recognition

Task

Identify text segments expressing references to named entities (NE) including

- person names
- company/organization names
- locations

and also

- dates&times
- percentages
- monetary amounts

Example of NE - annotated text

Delimit the named entities in a text and tag them with NE types:

```
<ENAMEX TYPE="LOCATION">Italy</ENAMEX>‘s business world was rocked by
the announcement <TIMEX TYPE="DATE">last Thursday</TIMEX> that Mr.
<ENAMEX TYPE="PERSON">Verdi</ENAMEX> would leave his job as vice-
president of <ENAMEX TYPE="ORGANIZATION">Music Masters of Milan,
Inc</ENAMEX> to become operations director of
<ENAMEX TYPE="ORGANIZATION">Arthur Andersen</ENAMEX>.
```

- “Milan” is part of organization name
- “Arthur Andersen” is a company
- “Italy” is a location

NE and Question - Answering

Often, the expected answer type of a question is a NE

- *What is the name of the first russian cosmonaut to do a walkspace?*
  Expected answer type is PERSON
- *Name the 5 most important software companies?*
  Expected answer type is a list of COMPANY
- *Where does the storming of the Bastille took place?*
  Expected answer type is LOCATION (subtype COUNTRY or TOWN)
- *When does the Storming of the Bastille took place?*
  Expected answer type is DATE

NER answers the questions: WHO, WHAT, WHEN, WHERE.

L I G

NER Challenges

- Potential set of NE is too numerous to include in dictionaries
- Names changing constantly
- Names appear in many variant forms
- Subsequent occurrences of names might be abbreviated (e.g., coreferences)

Simple search doesn't work well
Modern methods use context-based methods

Difficulties for Pattern Matching Approach

Whether a phrase is a named entity, and what name class it has, depends on

- Internal structure:
  "Mr. Brandon"
- Context:
  "The new company , SafeTek , will make air bags."
  "Augusta Ada King, Countess of Lovelace was an English mathematician and writer"

Annotation BIO coding

	IO encoding	BIO encoding	OIBES (or BILOU)
Sue	PER	B-PER	S-PER
showed	O	O	O
Steven	PER	B-PER	S-PER
Yann	PER	B-PER	B-PER
LeCun	PER	I-PER	E-PER
's	O	O	O
book	O	O	O
.	O	O	O

NER – Evaluation

NE

- F1-score$= 2 \times \frac{Precision \times Recall}{Precision + Recall}$
- sometimes accuracy or micro F1-score is used

State-of-the-art performance – dataset

The CoNLL 2003 NER task [Sang and De Meulder, 2003] consists of newswire text from the Reuters RCV1 corpus tagged with four different entity types (PER, LOC, ORG, MISC).
clips.uantwerpen.be/conll2003/ner/

```
U.N.         NNP  I-NP  I-ORG
official     NN   I-NP  O
Ekeus        NNP  I-NP  I-PER
heads        VBZ  I-VP  O
for          IN   I-PP  O
Baghdad      NNP  I-NP  I-LOC
.            .    O     O
```

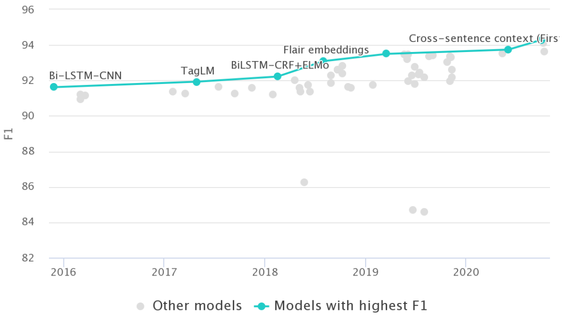
English data	Articles	Sentences	Tokens	LOC	MISC	ORG	PER
Training set	946	14,987	203,621	7140	3438	6321	6600
Development set	216	3,466	51,362	1837	922	1341	1842
Test set	231	3,684	46,435	1668	702	1661	1617

2003 CoNLL 2003 NER English results

FIJZ03 used an ensemble of Rule-based, HMM and MaxEnt classifers.

English	precision	recall	F
[FIJZ03]	88.99%	88.54%	88.76
[CN03]	88.12%	88.51%	88.31
[KSNM03]	85.93%	86.21%	86.07
[ZJ03]	86.13%	84.88%	85.50
[CMP03b]	84.05%	85.96%	85.00
...			
[HV03]	76.33%	80.17%	78.20
[DD03]	75.84%	78.13%	76.97
[Ham03]	69.09%	53.26%	60.15
baseline	71.91%	50.90%	59.61

2020 CoNLL 2003 NER English results



Extracted from paperswithcode.com

Most architectures use memory RNN + pre-trained embedding

NER through machine learning

1. Get a training dataset
2. Label tokens with its entity class or other (O)
3. Extract features for the task
4. Train a sequence classifier
5. Evaluate on a separate test set

Features for sequence labeling

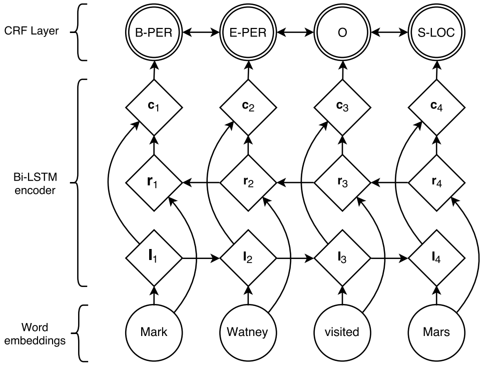
- Words (current, previous/next)
- Inferred linguistic features (e.g. POS)
- Label context (previous/next label)

Neural Architectures for Named Entity Recognition

Neural Architectures for Named Entity Recognition [Lample et al., 2016]

- Trained character-based word representations + pre-trained embeddings
- Bidirectional-LSTM encoder
- CRF decoder

Architecture



$\mathbf{l}_i$ represents the word i and its left context, $\mathbf{r}_i$ represents the word i and its right context. $\mathbf{c}_i$ is the concatenation of these two vectors.

Long Short-Term Memory (LSTM)

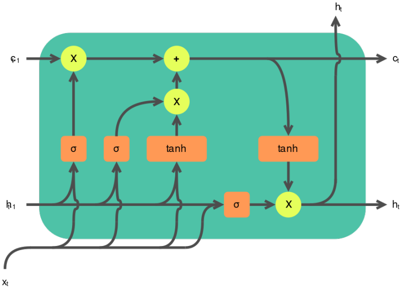
- Recurrent Neural Network works on sequential data
- RNNs fail to learn long dependencies and tend to be biased towards their most recent inputs in the sequence [Bengio et al., 1994]
- LSTMs uses a memory-cell to capture long-range dependencies through the use of several gates that control the proportion of the input to give to the memory cell, and the proportion from the previous state to forget [Hochreiter and Schmidhuber, 1997].

Long short-term memory (LSTM)

An LSTM unit is composed:

- a memory cell,
- an *input gate*,
- an *output gate* and
- a *forget gate*.

The cell stores values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.



(figure: wikipedia)

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) & (1)\\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) & (2)\\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) & (3)\\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) & (4)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t & (5)\\
h_t &= o_t \odot \sigma_h(c_t) & (6)
\end{aligned}
$$

$\odot$ is the element-wise product (Hadamard product).

L I G

Bidirectional LSTM (BiLSTM)

For a $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ a bi-LSTM

- computes a representation $\overrightarrow{\mathbf{h}_t}$ of the left context of the sentence at every word t.
- computes a representation of the right context $\overleftarrow{\mathbf{h}_t}$ using a second LSTM that reads the same sequence in reverse.

These are called the forward LSTM and the backward LSTM and are referred to as a bidirectional LSTM [Graves and Schmidhuber, 2005].

The output vector $\mathbf{h}_t$ is the concatenation of left and right context representations, $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$.

Sequence classification throught Conditional Random Fields

For an input sentence

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n), \text{ we have } \mathbf{P} = biLSTM(\mathbf{X})$$

$\mathbf{P}$ is of size $n \times k$, where k is the number of distinct tags, and $P_{i,j}$ corresponds to the score of the j^{th} tag of the i^{th} word in a sentence. For a sequence of predictions $\mathbf{y} = (y_1, y_2, \ldots, y_n)$

them using a conditional random field approach [Lafferty et al., 2001], the score s is computed as

$$s(\mathbf{X}, \mathbf{y}) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

where $\mathbf{A}$ is a matrix of transition scores such that $A_{i,j}$ represents the score of a transition from the tag i to tag j. y_0 and y_n are the *start* and *end* tags of a sentence, that we add to the set of possible tags.

L I G

Sequence classification throught Conditional Random Fields

A softmax over all possible tag sequences yields a probability for the sequence $\mathbf{y}$:

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{s(\mathbf{X},\mathbf{y})}}{\sum_{\widetilde{\mathbf{y}} \in \mathbf{Y_X}} e^{s(\mathbf{X},\widetilde{\mathbf{y}})}}.$$

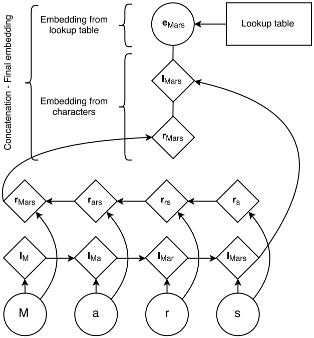
The training consists in maximizing the log-probability of the correct tag sequence:

$$\log(p(\mathbf{y}|\mathbf{X})) = s(\mathbf{X},\mathbf{y}) - \log\left(\sum_{\widetilde{\mathbf{y}} \in \mathbf{Y_X}} e^{s(\mathbf{X},\widetilde{\mathbf{y}})}\right) \tag{7}$$

where $\mathbf{Y_X}$ represents all possible tag sequences. The output sequence is the one that obtains the maximum score given by:

$$\mathbf{y}^* = \underset{\widetilde{\mathbf{y}} \in \mathbf{Y_X}}{\arg\max}\, s(\mathbf{X},\widetilde{\mathbf{y}}). \tag{8}$$

Features - character-level words + word embeddings



The character embeddings of the word "Mars" are given to a biLSTM. The left and right outputs are concatenated with the word embedding to obtain a representation for this word.

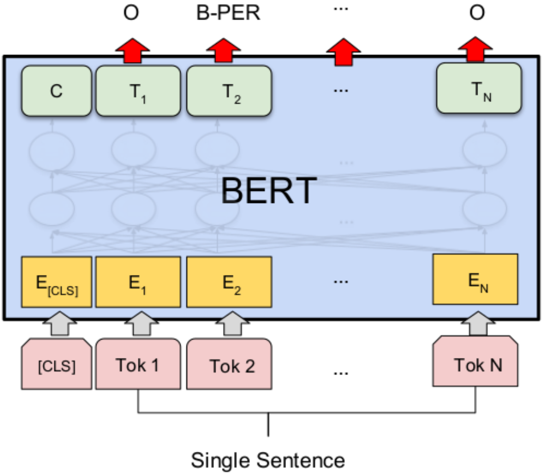
Results

Model	F_1
Collobert et al. (2011) *	89.59
Lin and Wu (2009)*	90.90
Huang et al. (2015)*	90.10
Passos et al. (2014)*	90.90
Luo et al. (2015)*	**91.2**
Chiu and Nichols (2015)*	90.77
LSTM-CRF (no char)	90.20
LSTM-CRF	**90.94**

English NER results (CoNLL-2003 test set).
indicates models trained with the use of external labelled data

NER throught BERT fine tuning

Keep the same architecture as original BERT but change the task [Devlin et al., 2019]



Single Sentence Tagging CoNLL-2003 NER

BERT fine tunning results

System	Dev F1	Test F1
ELMo [Peters et al., 2018]	95.7	92.2
CVT [Clark et al., 2018]	-	92.6
CSE [Akbik et al., 2018]	-	**93.1**
BERT Fine-tuning approach		
BERT_{large}	96.6	92.8
BERT_{base}	96.4	92.4

Natural Language Understanding

NLU as sequence labelling task

Named Entity Recognition (NER)

Slot Filling

Slot-filling

Sentence	*show*	*flights*	*from*	*Boston*	*To*	*New*	*York*	*today*
Slots/Concepts	O	O	O	B-dept	O	B-arr	I-arr	B-date
Named Entity	O	O	O	B-city	O	B-city	I-city	O
Intent	*Find_Flight*							
Domain	*Airline Travel*							

- Intent recognition influenced by speech act theory [Searle and Searle, 1969]
  $\rightarrow$ often addressed as a classification problem
- Slot/Concept influenced by Frame semantics [Fillmore et al., 1976]
  $\rightarrow$ often addressed as a sequence labelling problem

Speech Acts (aka Dialogue Acts)

Speech acts [Searle and Searle, 1969, Bach and Harnish, 1979]

Intent recognition

- Constatives: committing the speaker to something's being the case
  answering, claiming, confirming, denying, disagreeing, stating
- Directives: attempts by the speaker to get the addressee to do something
  advising, asking, forbidding, inviting, ordering, requesting
- Commissives: committing the speaker to some future course of action
  promising, planning, vowing, betting, opposing
- Acknowledgments: express the speaker's attitude regarding the hearer with respect to some social action
  apologizing, greeting, thanking, accepting an acknowledgment

Speech acts: examples

```
"Turn up the music!"
Directive

"What day in May do you want to travel?"
Directive

"I need to travel in May"
Constative

"Thanks!"
Acknowledgement
```

The Frame: getting the content

A set of slots, to be filled with information of a given type.

Each associated with a question to the user

Slot	Type	Question
ORIGIN	city	"What city are you leaving from?"
DEST	city	"Where are you going?"
DEP_DATE	date	"What day would you like to leave?"
DEP_TIME	time	"What time would you like to leave?"
AIRLINE	line	"What is your preferred airline?"

The slot-filling approach: Industry

	On-device solution	In-house server hosting	Third-party server hosting	Private mode on third-party server?	Built-in intents	Custom intents
Alexa	✖ No	✖ No	✔ Yes	✖ No	✔ Yes	✔ Yes
Api	✖ No	✖ No	✔ Yes	✔ Yes	✔ Yes	✔ Yes
Luis	✖ No	✖ No	✔ Yes	✖ No	✔ Yes	✔ Yes
Siri	✖ No	✖ No	✔ Yes	✖ No	✔ Yes	✖ No
Snips	✔ Yes	✔ Yes	✖ No	N/A	✔ Yes	Soon
Watson	✖ No	✖ No	✔ Yes	✖ No	✖ No	✔ Yes

from Snips Blog `https://medium.com/snips-ai`

plus other dialogue frameworks including NLU such as Rasa (`https://rasa.com/`), ParlAI (`https://parl.ai/`), Botkit (`https://botkit.ai/`)

→ most working on textual inputs (or transcripts).
→ fluent.ai supposed to do E2E intent recognition.

The slot-filling approach: Evaluation

Intent recognition

- F1-score$= 2 \times \frac{Precision \times Recall}{Precision + Recall}$
- sometimes accuracy or micro F1-score is used

Slot/Value Recognition

- F1-score$= 2 \times \frac{Precision \times Recall}{Precision + Recall}$
- Concept-Error-Rate (CER) or slot-value (CVER) level [Chotimongkol and Rudnicky, 2001]

Task completion success

to which extend the system enables the user to achieve a task?
use of objective metrics (e.g., success rate, completion time)
and subjective metrics (e.g., questionnaire)

Concept Error Rate

Concept Error Rate (CER)
[Boros et al., 1996, Chotimongkol and Rudnicky, 2001] or slot error rate (SER).

$$CER = 100 \left(\frac{SU_S + SU_I + SU_D}{SU} \right) \% \tag{9}$$

SU is the total number of semantic units in the reference answer and SU_S, SU_I, and SU_D are the number of semantic units that were substituted, inserted, and deleted.

Spoken:	No	to Bonn
REF:	`dm_marker:no`	`goalcity:Bonn`
Recog.:	No	to Berlin
HYP:	`dm_marker:no`	`goalcity:Berlin`

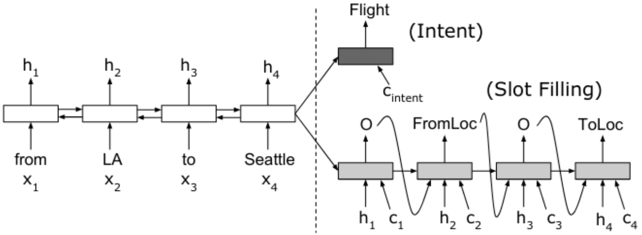
can be done at the concept/slot only or for the couple Concept/Value. It is called Concept Value Error Rate (CVER) in that case.

NLU – multitask sequence labelling

State-of-the-art NLU

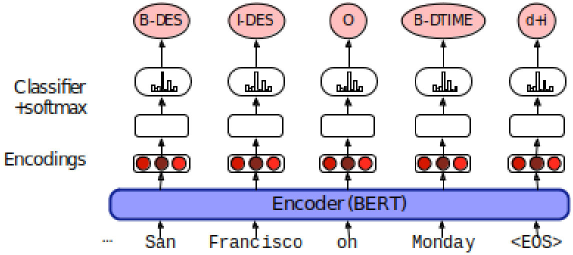
- CRF models [Jeong and Lee, 2008]
- DNN-based models [Mesnil et al., 2015, Bapna et al., 2017, Liu and Lane, 2016, Huang et al., 2017]

.



from [Liu and Lane, 2016]

Slot filling using contextual embeddings



Can do domain and intent too: e.g., generate the label "AIRLINE_TRAVEL + SEARCH_FLIGHT"

Alignement constraint

- *BIO* NE labeling scheme → very efficient results
- Cannot be assumed in an E2E context
- Prevent abstraction

NLU – generation approach

Task with unaligned data

Seq2seq (“turn on the light")
`(Source) allume la lumière`
`(Target) intent[set_device], action[turn on], device[light]`

Advantages

- Abstraction made possible
- A single model for all tasks (intent/slot/value)

NLU – results

Sequence labelling and generation performances on the VocADom@A4H [Desot et al., 2019]

NLU Model +Data set	Intent F1-score	Slot F1-score
Aligned:		
Rasa-NLU	76.57	**79.03**
Tri-CRF	76.36	60.64
Att-RNN	**96.70**	74.27
Unaligned:		
Seq2seq1	**94.74**	51.06
Seq2seq2	85.51	**65.49**

Seq2seq1 Same training corpus

Seq2seq2 extra slots considered (from Elso)

Summary

NLU covers a wide range of tasks

NER and slot-filling are frequently approached as

- Classification task (e.g., intent)
- Sequence labelling task (e.g., NER, slot-filling)
- Less frequently as a generation task

Use of pre-trained models (e.g., ELMO, BERT etc) brought a clear improvement.

A lot to be done before reaching a real understanding (common grounding, common sense, etc.)

A lot of work on knowledge extraction

L I G

References I

Akbik, A., Blythe, D., and Vollgraf, R. (2018).
Contextual string embeddings for sequence labeling.
In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Bach, K. and Harnish, R. (1979).
Linguistic communication and speech acts.
Cambridge: MIT Press.

Bahdanau, D., Cho, K., and Bengio, Y. (2015).
Neural machine translation by jointly learning to align and translate.

Bapna, A., Tur, G., Hakkani-Tur, D., and Heck, L. (2017).
Towards zero-shot frame semantic parsing for domain scaling.
arXiv:1707.02363 [cs].

Bengio, Y., Simard, P., and Frasconi, P. (1994).
Learning long-term dependencies with gradient descent is difficult.
IEEE Transactions on Neural Networks, 5(2):157–166.

Bobrow, D. G. (1964).
Natural language input for a computer problem solving system.
Technical report, USA.

Boros, M., Eckert, W., Gallwitz, F., Gorz, G., Hanrieder, G., and Niemann, H. (1996).
Towards understanding spontaneous speech: word accuracy vs. concept accuracy.
In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 2, pages 1009–1012 vol.2.

References II

Chotimongkol, A. and Rudnicky, A. I. (2001).
N-best speech hypotheses reordering using linear regression.
In *Seventh European Conference on Speech Communication and Technology*.

Clark, K., Luong, M.-T., Manning, C. D., and Le, Q. (2018).
Semi-supervised sequence modeling with cross-view training.
In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925.

Desot, T., Portet, F., and Vacher, M. (2019).
SLU FOR VOICE COMMAND IN SMART HOME: COMPARISON OF PIPELINE AND END-TO-END APPROACHES.
In *IEEE Automatic Speech Recognition and Understanding Workshop*, Sentosa, Singapore, Singapore.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019).
BERT: Pre-training of deep bidirectional transformers for language understanding.
In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fillmore, C. J. et al. (1976).
Frame semantics and the nature of language.
In *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*, volume 280, pages 20–32.

References III

Graves, A. and Schmidhuber, J. (2005).
Framewise phoneme classification with bidirectional lstm and other neural network architectures.
Neural Networks, 18(5):602–610.

Hemphill, C. T., Godfrey, J. J., and Doddington, G. R. (1990).
The atis spoken language systems pilot corpus.
In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Hochreiter, S. and Schmidhuber, J. (1997).
Long short-term memory.
Neural computation, 9(8):1735–1780.

Huang, L., Sil, A., Ji, H., and Florian, R. (2017).
Improving slot filling performance with attentive neural networks on dependency structures.
arXiv:1707.01075 [cs].

Jeong, M. and Lee, G. G. (2008).
Triangular-chain conditional random fields.
IEEE Transactions on Audio, Speech, and Language Processing, 16(7):1287–1302.

Jurafsky, D. and Manning, C. (2021).
Speech and Language processing.
Pearson New International Edition.

References IV

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001).
Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
In *International Conference on Machine Learning,ICML'01*, pages 282–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016).
Neural architectures for named entity recognition.
In *Proceedings of NAACL-HLT*, pages 260–270.

Liu, B. and Lane, I. (2016).
Attention-based recurrent neural network models for joint intent detection and slot filling.
In *Proceedings of Interspeech 2016*, pages 685–689.

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., and others (2015).
Using recurrent neural networks for slot filling in spoken language understanding.
IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 23(3):530–539.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018).
Deep contextualized word representations.
In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

References V

Sang, E. T. K. and De Meulder, F. (2003).
Introduction to the conll-2003 shared task: Language-independent named entity recognition.
In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Searle, J. R. and Searle, J. R. (1969).
Speech acts: An essay in the philosophy of language, volume 626.
Cambridge university press.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014).
Sequence to sequence learning with neural networks.
In *NIPS2014*.

Williams, J., Raux, A., and Henderson, M. (2016).
The dialog state tracking challenge series: A review.
Dialogue & Discourse, 7(3):4–33.

Winograd, T. (1970).
Procedures as a representation for data in a computer program for understanding natural language.
PhD thesis, Massachusetts Institute of Technology.