Character-based Neural Semantic Parsing

What did we learn and where do we go next?

Rik van Noord

PhD-thesis 2016-2020

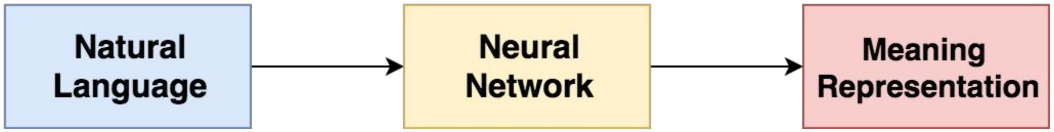
Advisors: Johan Bos and Antonio Toral

Goal: Create neural methods to produce structured meaning representations

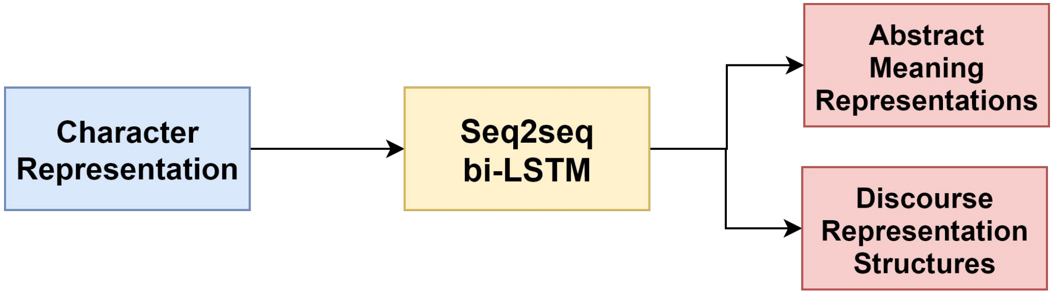
Character-based Neural Semantic Parsing

- Traditional semantic parsing

- Neural semantic parsing

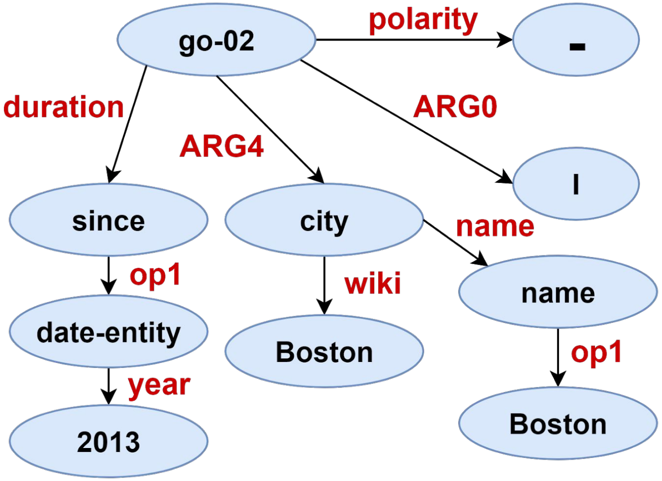


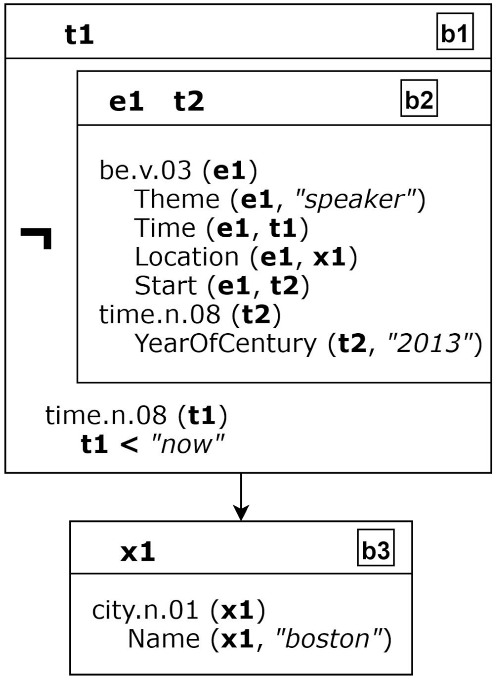
- Our work: character-based neural semantic parsing



Meaning representations

AMR **I haven't been to Boston since 2013.** **DRS**

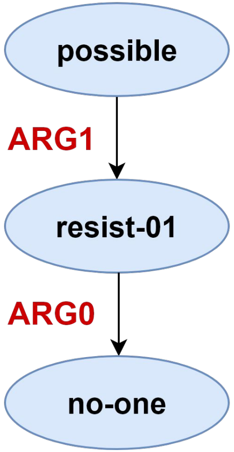




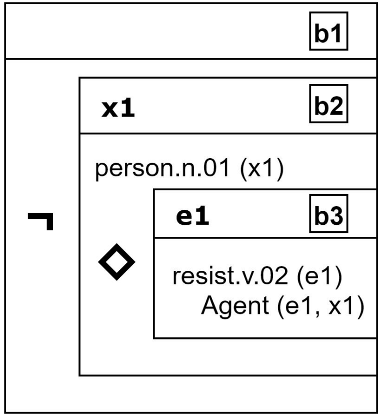
Meaning representations

No one could resist.

AMR



DRS



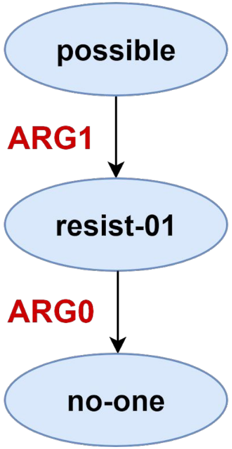
Text-based meaning representations

No one could resist.

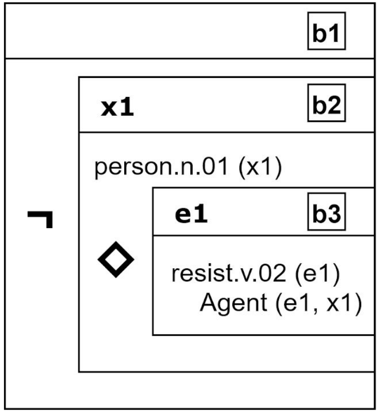
Text-AMR

```
( p / possible
    :ARG1 (r / resist-01
        :ARG0 (n / no-one)))
```

AMR



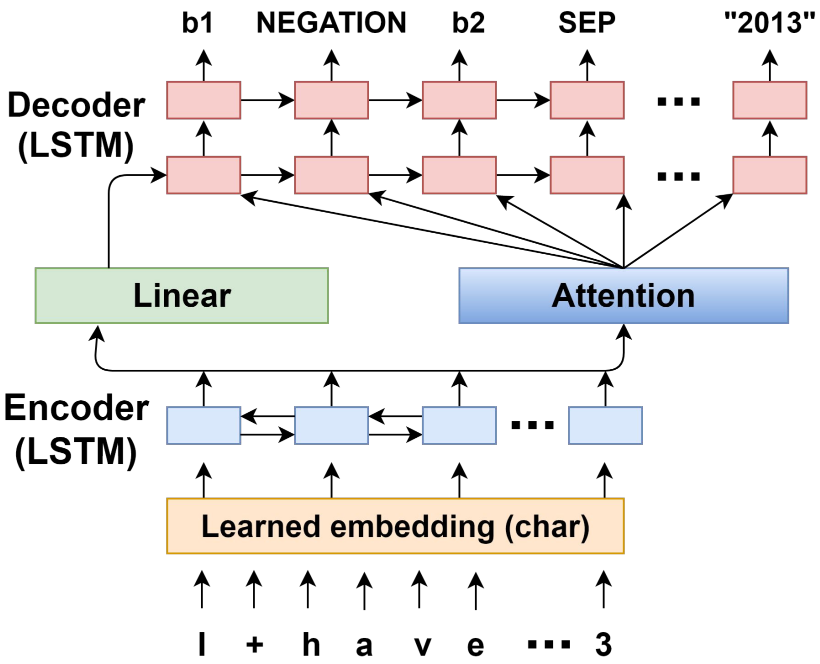
DRS



Text-DRS

b1 NOT b2
b2 REF x1
b2 person "n.01" x1
b2 POS b3
b3 REF e1
b3 Agent e1 x1
b3 resist "v.02" e1

Sequence-to-sequence model



Character-level models

Input: ^ i + h a v e n ‘ t + b e e n + t o +
^ b o s t o n + s i n c e + 2 0 1 7 .

AMR: (p o s s i b l e + :ARG1 + (r e s i s t - 0 1
+ :ARG0 + (n o - o n e)))

DRS: b1 NOT b2 *** b2 REF x1 *** b2 + p e r s o n +
“ n . 0 1 “ + x1 *** b2 POS b3 *** b3 REF e1 *** b3
Agent e1 x1 *** b3 + r e s i s t + “ v . 0 2 “ + e1

Data sets

- Semantic parsing data sets are quite small
- **AMR:** LDC2017T10 with 36,521 gold standard examples
- **DRS:** PMB release 2.1.0 with 3,998 gold standard examples
    - Experiments only on English - for data see pmb.let.rug.nl

Main Findings

Finding 1: Character-level models work surprisingly well!

Outperformed word-level models for both AMR and DRS

Also outperformed BPE for DRS parsing

Takeaway: characters can be an interesting baseline

Why do characters work so well?

- No assumptions fed to the model
- Can deal with spelling errors
- Can model and learn inflections
- Can deal with unknown words

Why do characters work so well?

- No assumptions fed to the model
- Can deal with spelling errors
- Can model and learn inflections
- Can deal with unknown words

But if this was universally true, everybody would be using characters already

Why do characters work so well?

- No assumptions fed to the model
- Can deal with spelling errors
- Can model and learn inflections
- Can deal with unknown words

But if this was universally true, everybody would be using characters already

- Impossible to learn large vocabularies for small data sets
- At least for characters, we get to do lots of updates for each character

Main Findings

Finding 2: It helps to rewrite variables to a different representation

Original:

b1 NOT b2
b2 REF x1
b2 person "n.01" x1
b2 POS b3
b3 REF e1
b3 Agent e1 x1
b3 resist "v.02" e1

Main Findings

Finding 2: It helps to rewrite variables to a different representation

Original:

b1 NOT **b2**
b2 REF **x1**
b2 person "n.01" **x1**
b2 POS **b3**
b3 REF **e1**
b3 Agent **e1** **x1**
b3 resist "v.02" **e1**

Main Findings

Finding 2: It helps to rewrite variables to a different representation

Original:

b1 NOT b2
b2 REF x1
b2 person "n.01" x1
b2 POS b3
b3 REF e1
b3 Agent e1 x1
b3 resist "v.02" e1

Absolute:

$0 NOT $1
$1 REF @1
$1 person "n.01" @1
$1 POS $2
$2 REF @2
$2 Agent @2 @1
$2 resist "v.02" @2

Main Findings

Finding 2: It helps to rewrite variables to a different representation

Original:

b1 NOT **b2**
b2 REF **x1**
b2 person "n.01" **x1**
b2 POS **b3**
b3 REF **e1**
b3 Agent **e1** **x1**
b3 resist "v.02" **e1**

Absolute:

$0 NOT **$1**
$1 REF **@1**
$1 person "n.01" **@1**
$1 POS **$2**
$2 REF **@2**
$2 Agent **@2** **@1**
$2 resist "v.02" **@2**

Relative:

b1 NOT b2
b2 REF x1
b2 person "n.01" x1
b2 POS b3
b3 REF e1
b3 Agent e1 x1
b3 resist "v.02" e1

Main Findings

Finding 2: It helps to rewrite variables to a different representation

Original:

b1 NOT b2
b2 REF x1
b2 person "n.01" x1
b2 POS b3
b3 REF e1
b3 Agent e1 x1
b3 resist "v.02" e1

Absolute:

$0 NOT $1
$1 REF @1
$1 person "n.01" @1
$1 POS $2
$2 REF @2
$2 Agent @2 @1
$2 resist "v.02" @2

Relative:

$NEW NOT $NEW
$0 REF @NEW
$0 person "n.01" @0
$0 POS $NEW
$0 REF @NEW
$0 Agent @0 @-1
$0 resist "v.02" @0

Main Findings

Finding 2: It helps to rewrite variables to a different representation

Original:	Absolute:	Relative:
b1 NOT b2	$0 NOT $1	$NEW NOT $NEW
b2 REF x1	$1 REF @1	$0 REF @NEW
b2 person "n.01" x1	$1 person "n.01" @1	$0 person "n.01" @0
b2 POS b3	$1 POS $2	$0 POS $NEW
b3 REF e1	$2 REF @2	$0 REF @NEW
b3 Agent e1 x1	$2 Agent @2 @1	$0 Agent @0 @-1
b3 resist "v.02" e1	$2 resist "v.02" @2	$0 resist "v.02" @0

Takeaway: Important to take care of your variables

Main Findings

Finding 3: using silver data improves performance a lot

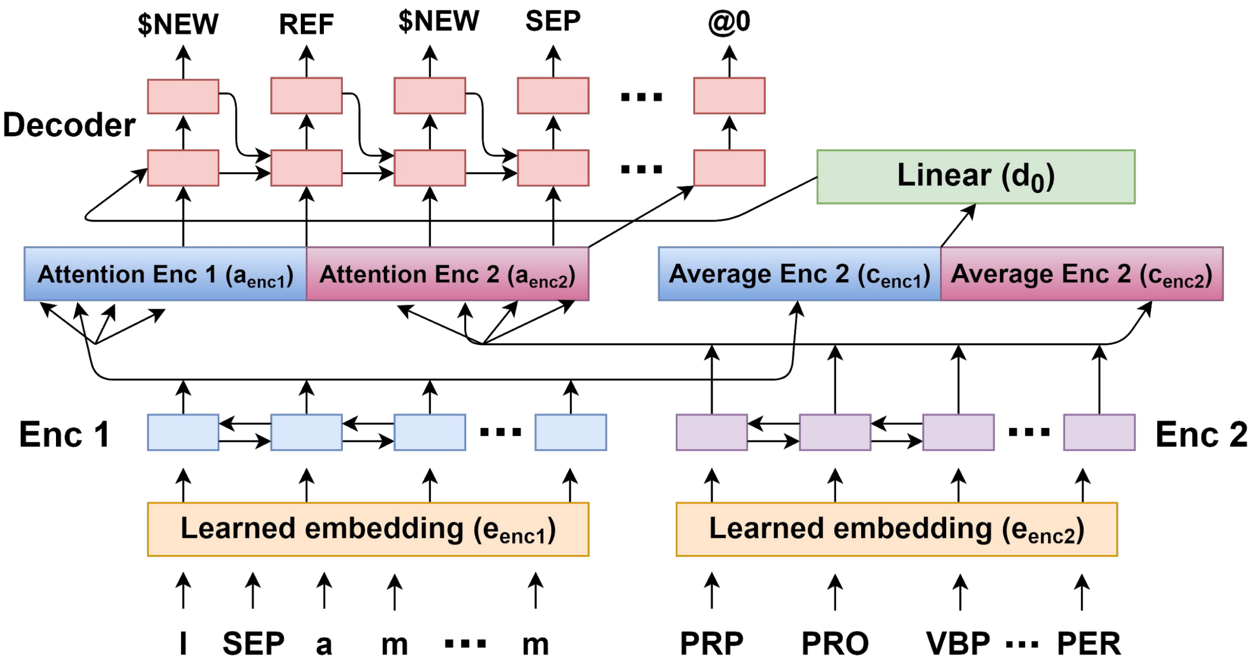
- Self-training or using a different parser
- Pretrain on gold + silver, finetune on gold
- There is a limit to improved performance

For **AMR:** F-score of 64 to 71

For **DRS:** F-score of 78 to 84

Main Findings

Idea: improve performance using linguistic features



Main Findings

	Gold
Baseline	78.6
+ LEM	+1.3
+ SEM	+1.9
+ POS	+2.2
+CCG	+2.4
+DEP	+2.7

Main Findings

	Gold	Gold + silver
Baseline	78.6	84.5
+ LEM	+1.3	+1.1
+ SEM	+1.9	+1.0
+ POS	+2.2	+1.1
+CCG	+2.4	+0.9
+DEP	+2.7	+0.6

Main Findings

Takeaway: important to use silver data to create a strong baseline

	Gold	**Gold + silver**
Baseline	78.6	84.5
+ LEM	**+1.3**	**+1.1**
+ SEM	**+1.9**	**+1.0**
+ POS	**+2.2**	**+1.1**
+CCG	**+2.4**	**+0.9**
+DEP	**+2.7**	**+0.6**

Main Findings

Takeaway: important to use silver data to create a strong baseline

Unpublished result: tried multi-task learning with AMR and DRS

In short: always better to add silver data instead of data from the other formalism

But now there are pretrained language models!

Are characters still useful?

Paper in a nutshell

We know character-level representations did well on semantic parsing before pretrained LMs, but are they still useful now?

Yes!

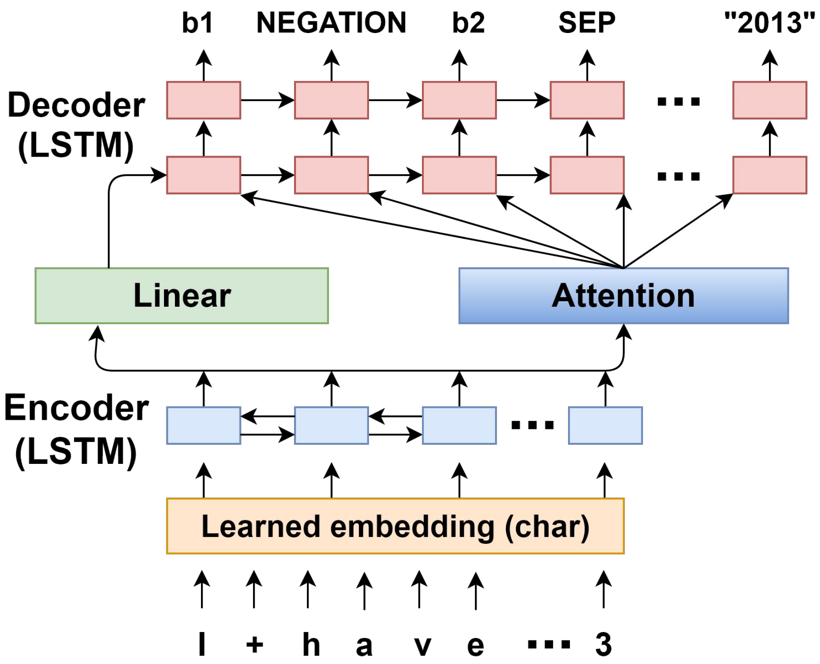
Paper in a nutshell

We know character-level representations did well on semantic parsing before pretrained LMs, but are they still useful now?

Yes!

- Two methods of **combining** characters with pretrained LMs
- Significant improvements for AMR and especially DRS parsing
- Robust across languages, language models and data sets

Original system



Baselines

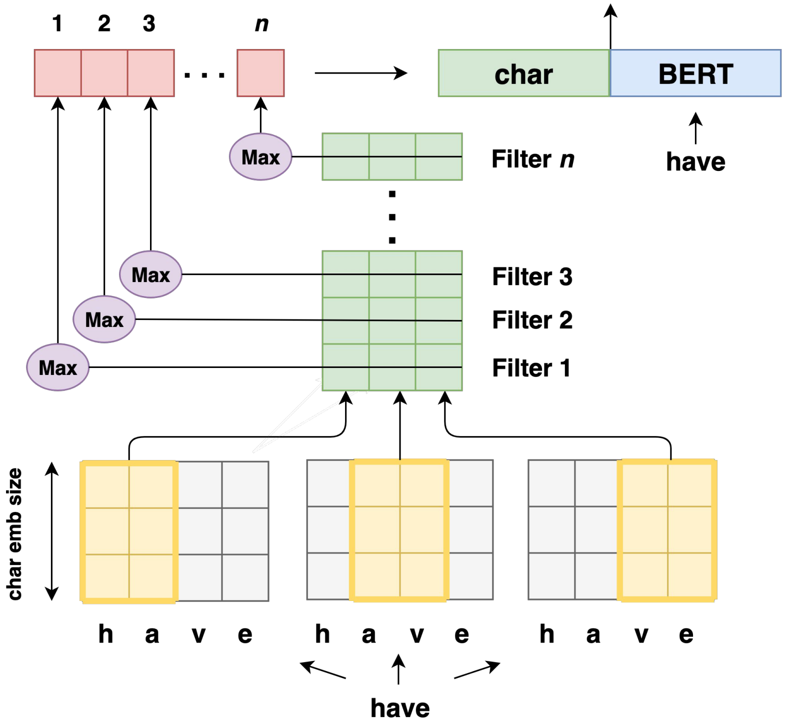
	DRS	AMR
Char	86.1	65.2
Word	85.3	62.2
ELMo	87.3	66.7
BERT-base	87.6	69.5
BERT-large	87.5	68.5
RoBERTa-base	87.0	67.1
RoBERTa-large	86.8	66.2

Baselines

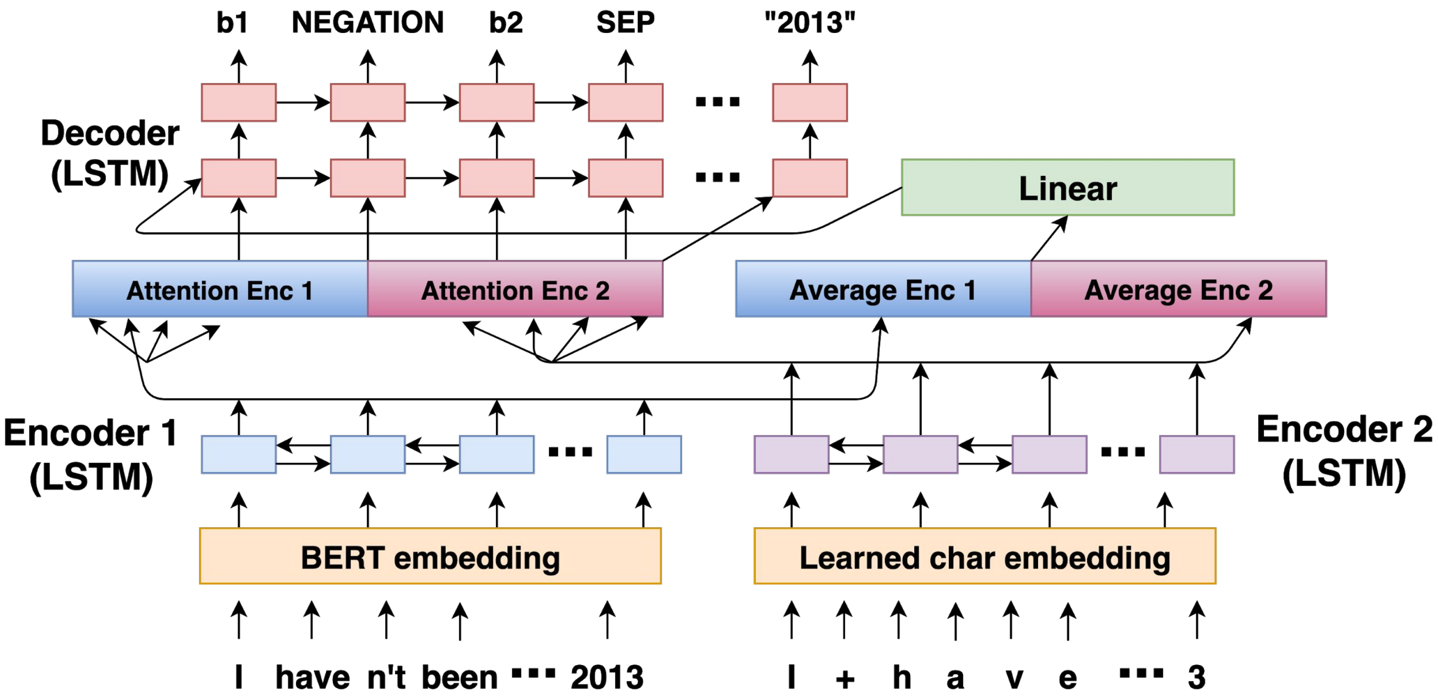
	DRS	AMR
Char	86.1	65.2
Word	85.3	62.2
ELMo	87.3	66.7
BERT-base	87.6	69.5
BERT-large	87.5	68.5
RoBERTa-base	87.0	67.1
RoBERTa-large	86.8	66.2

Can we combine characters with BERT?

Characters in one encoder: char-CNN



Characters in two encoders



Adding characters to BERT-base

	Baseline	+ char (1 enc)	+ char (2 enc)
DRS	87.6	88.1	88.1
AMR	70.5	71.0	70.4

All scores are averages of 5 runs

Adding characters to BERT-base

	Baseline	+ char (1 enc)	+ char (2 enc)
DRS	87.6	+0.5	+0.5
AMR	70.5	+0.5	-0.1

All scores are averages of 5 runs

Adding characters to LMs - DRS

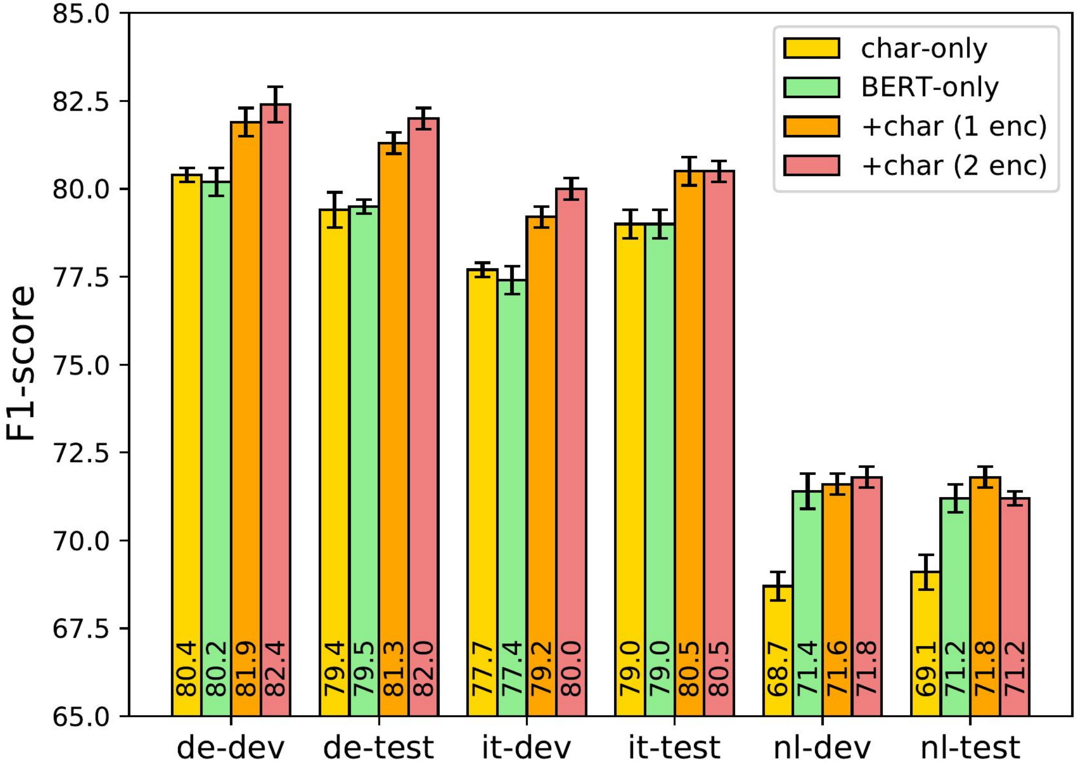
	Baseline	+ char (1 enc)	+ char (2 enc)
BERT-base	87.6	+0.5	+0.5
BERT-large	87.5	+0.7	+0.2
ELMo	87.3	+0.3	+0.5
RoBERTa-base	87.0	+0.3	+0.8
RoBERTa-large	86.8	+0.0	+0.2

What about other (linguistic) features?

BERT	87.6
BERT + char (1 enc)	+0.5
BERT + char (2 enc)	+0.5
BERT + GloVe	+0.3
BERT + FastText	+0.2
BERT + POS	+0.0
BERT + SEM	+0.3
BERT + LEM	+0.2
BERT + DEP	+0.3
BERT + CCG	+0.2

BERT	87.6
BERT + char (1 enc)	**+0.5**
BERT + char (2 enc)	**+0.5**
BERT + GloVe	+0.3
BERT + FastText	+0.2
BERT + POS	+0.0
BERT + SEM	+0.3
BERT + LEM	+0.2
BERT + DEP	+0.3
BERT + CCG	+0.2

What about other languages?



85.0
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
F1-score
char-only
BERT-only
+char (1 enc)
+char (2 enc)
80.4
80.2
81.9
82.4
79.4
79.5
81.3
82.0
77.7
77.4
79.2
80.0
79.0
79.0
80.5
80.5
68.7
71.4
71.6
71.8
69.1
71.2
71.8
71.2
de-dev
de-test
it-dev
it-test
nl-dev
nl-test

Main findings

Finding 1: Characters can be used in combination with LMs

Finding 2: They might be better than other extra resources

Finding 3: Improvements are robust across languages, LMs and formalisms

Main findings

Finding 1: Characters can be used in combination with LMs

Finding 2: They might be better than other extra resources

Finding 3: Improvements are robust across languages, LMs and formalisms

We should model characters in LMs, we just don’t know how to do it efficiently

CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters

Hicham El Boukkouri[1], Olivier Ferret[2], Thomas Lavergne[1], Hiroshi Noji[3], Pierre Zweigenbaum[1], Junichi Tsujii[3]

CANINE: Pre-training an Efficient Tokenization-Free Encoder for Language Representation

Jonathan H. Clark, Dan Garrette, Iulia Turc, John Wieting
Google Research

ByT5: Towards a token-free future with pre-trained byte-to-byte models

Linting Xue* Aditya Barua* Noah Constant* Rami Al-Rfou*
Sharan Narang Mihir Kale Adam Roberts Colin Raffel
Google Research

Chinese DRS parsing

- Work of PhD-student Chunliu Wang, will be presented at ACL 2021
- Interesting: Chinese characters already contain meaning
- Vocabulary is a lot larger, so closer to word-level

Findings are similar: characters outperform both word-level and BPE models

What’s next for semantic parsing?

We should use the meaning representations for something!

Future

Use semantic parsing for downstream applications

- Was always the goal of open domain semantic parsing
- English AMR parsing has higher scores than human agreement

Other potential use cases of semantic parsing

- Explainability - DRSs provide a grounded representation
- Evaluation - perhaps to give a general check of the semantics

What formalism has the most potential?

- LMs solve the easy cases, semantic parsing needed for the hard ones

Take-home message

Don't underestimate the power of characters!

Contact:

- @RikVanNoord
- www.rikvannoord.nl, rikvannoord@gmail.com

pmb.let.rug.nl

DRS-to-text generation

- Also work of Chunliu Wang, will present at GEM workshop 2021

- 5 DRS **challenge sets**
    - **Original** : Tom has three thousand books.
    - Tense : Tom had three thousand books.
    - Polarity : Tom does not have three thousand books.
    - Named entities : Kirk has three thousand books.
    - Grammatical num: Tom has one book.
    - Quantities : Tom has 3,200 books.
- New annotation/evaluation metric **ROSE** (Robust Overall Semantic Evaluation)
    - Semantics, Grammaticality, Phenomenon

DRS-to-text output examples

	Reference text	Generated text	Sem.	Gram.	Phen.	ROSE
(a)	She liked short skirts.	She liked short tomical.	0	0	1	0
(b)	Tom does not have three thousand books.	Tom never has three thousand books.	0	1	1	0
(c)	The small skirt will be pink.	The small skirt was pink.	0	1	0	0
(d)	He left 157 minutes ago.	He left fifteen minutes ago.	0	1	0	0
(e)	I checked it nine times.	I checked it nine.	0	0	1	0
(f)	We are painting the house green.	I paint the house green.	1	1	1	1
(g)	That hat cost around fifty dollars.	This hat cost about 50 dollars.	1	1	1	1
(h)	When I painted this picture, I was 23 years old.	I painted the picture when I was twenty-three years old.	1	1	1	1