Cross-Domain Semantic Parsing via Paraphrasing

Yu Su & Xifeng Yan, EMNLP 2017
presented by Sha Li

Semantic Parsing

Mapping **natural language utterances** to **logical forms** that machines can act upon.

which country had the highest carbon emissions last year

```
SELECT    country.name
FROM      country, co2_emissions
WHERE     country.id = co2_emissions.country_id
AND       co2_emissions.year = 2014
ORDER BY co2_emissions.volume DESC
LIMIT     1;
```

Database query

angelina jolie net worth

```
(FactoidQuery
  (Entity /m/0f4vbz)
  (Attribute /person/net_worth))
```

play sunny by boney m

```
(PlayMedia
  (MediaType MUSIC)
  (SongTitle "sunny")
  (MusicArtist /m/017mh))
```

Intents and arguments for a personal assistant

In-domain VS Cross-domain Semantic Parsing

- In-domain: training/test set from the same domain
- Cross-domain: train on **source domain** and test on **target domain**
- Why cross-domain:
  - Sometimes we have more training data from one domain than another; collecting training data from the target domain is expensive
  - The source domain shares some similarities with the target domain, making it possible to train a cross-domain model

Challenges

1. Different domains have different logical forms (different predicate names etc.) ⇒ translate to a common middle ground: **canonical utterance**

Canonical utterance: has a one-to-one mapping to the logical form

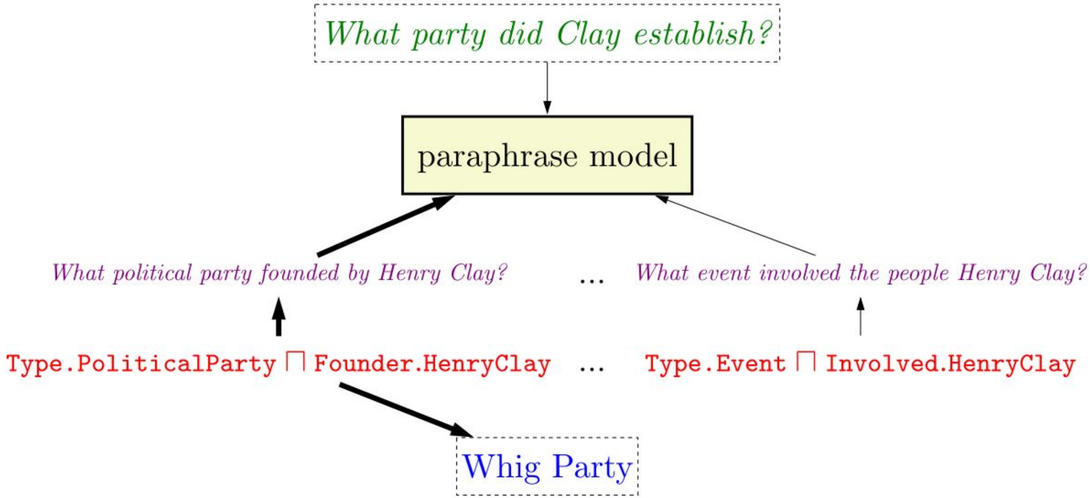
2. Vocabulary gap between domains ⇒ pretrained **word embeddings**

45%-70% of the words are covered by any of the other domains

Previous Work

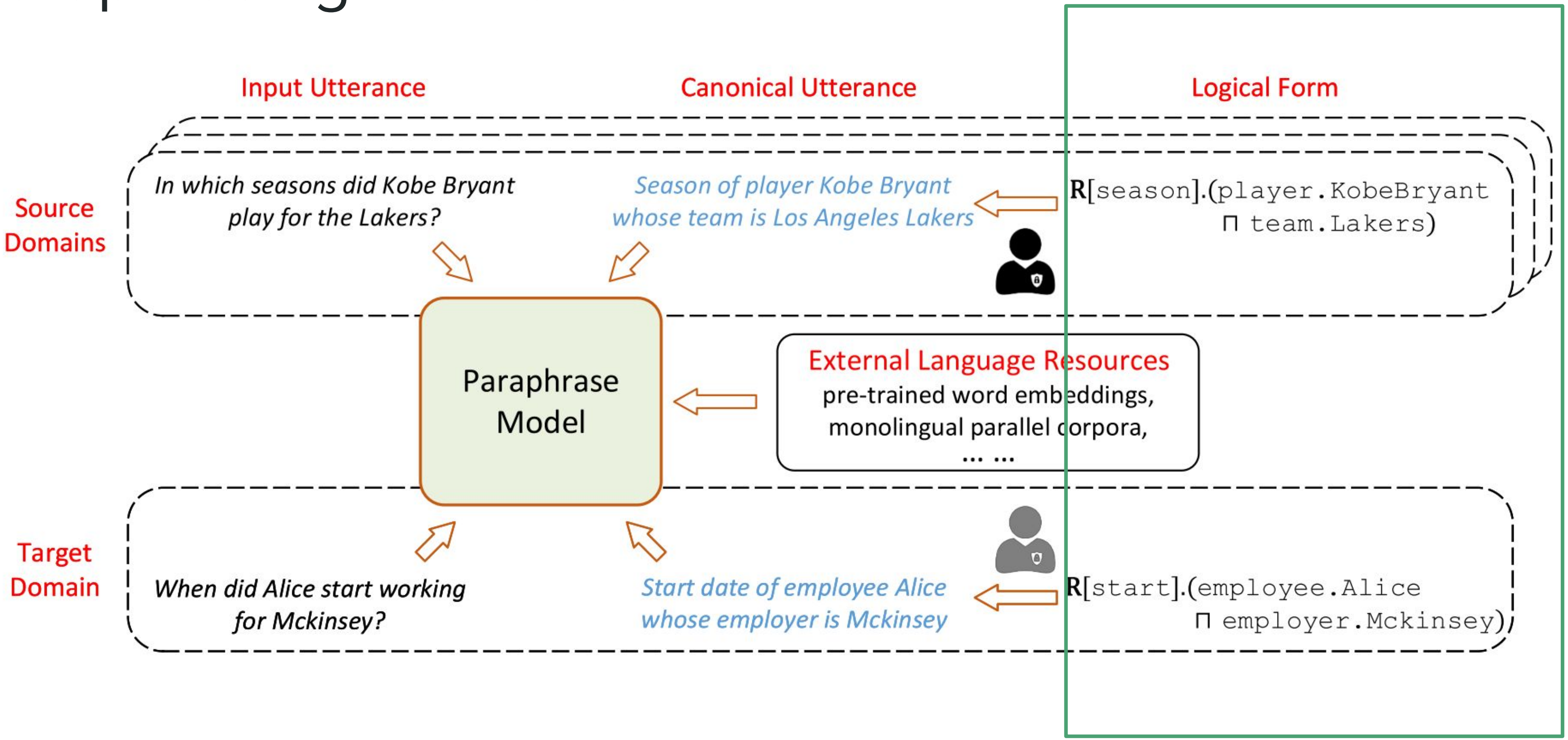
Paraphrase based semantic parsing

Map utterances into a canonical natural language form before transforming into logical form. (Berant and Liang 2014, Wang et al. 2015)



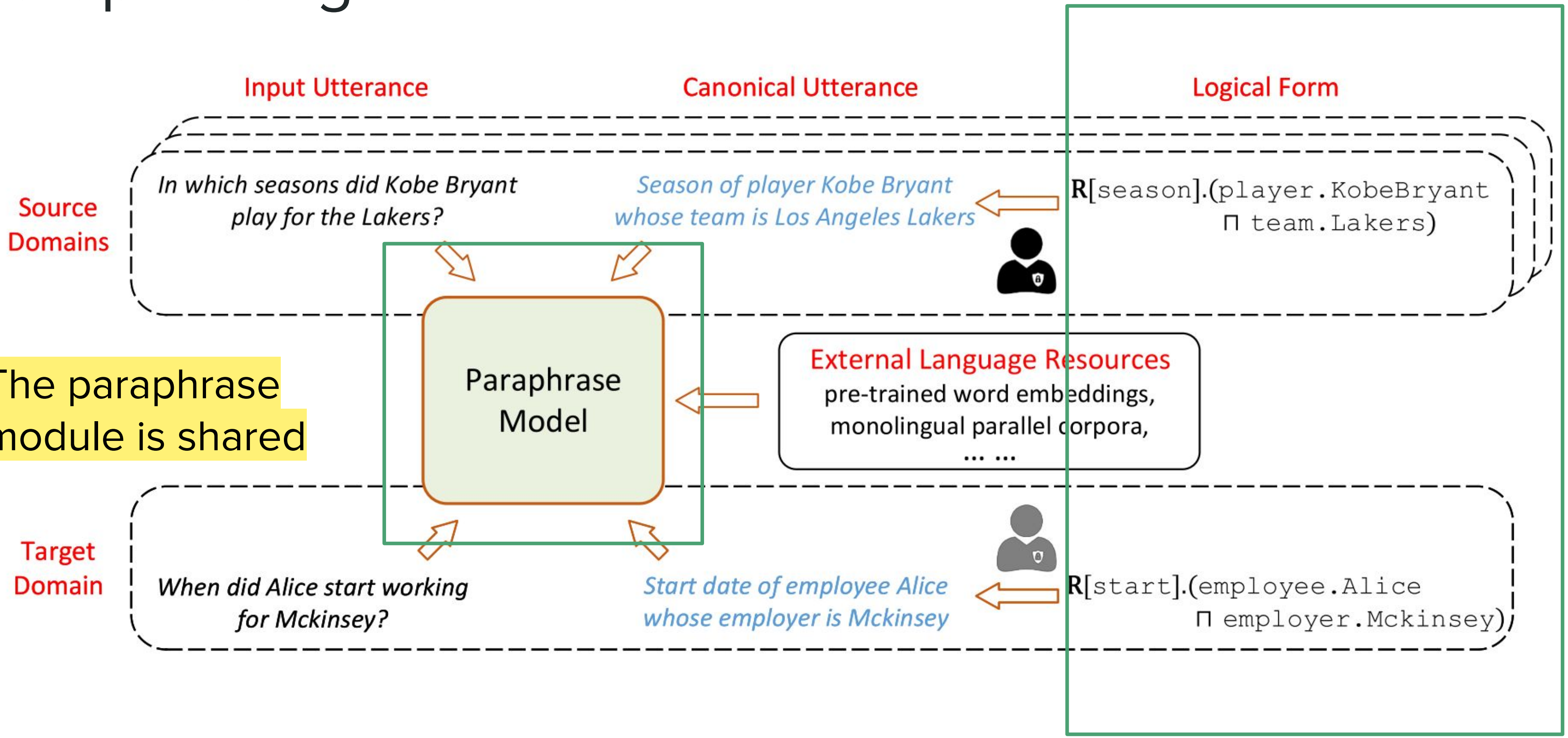
Paraphrasing Framework

The logical form is not shared across domains



The logical form is not shared across domains

Paraphrasing Framework



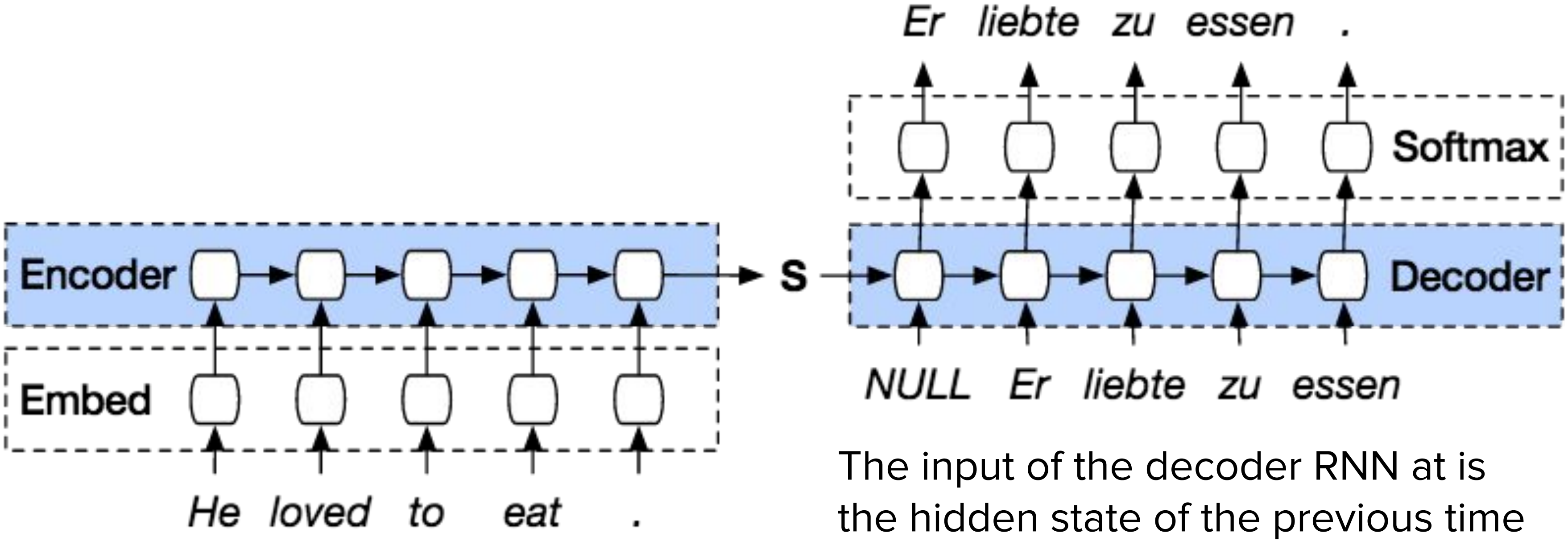
The paraphrase module is shared

Problem Setting

- Assume that the mapping from canonical utterance to logical form is given for both domains
- Propose a seq2seq model for **paraphrasing**
- Use **pre-trained word embeddings** to help domain adaptation
    - Introduce standardization techniques to improve word embeddings
- **Domain adaptation** is done by: training a paraphrase model in the source domain and fine-tuning it the target domain

Paraphrase Model

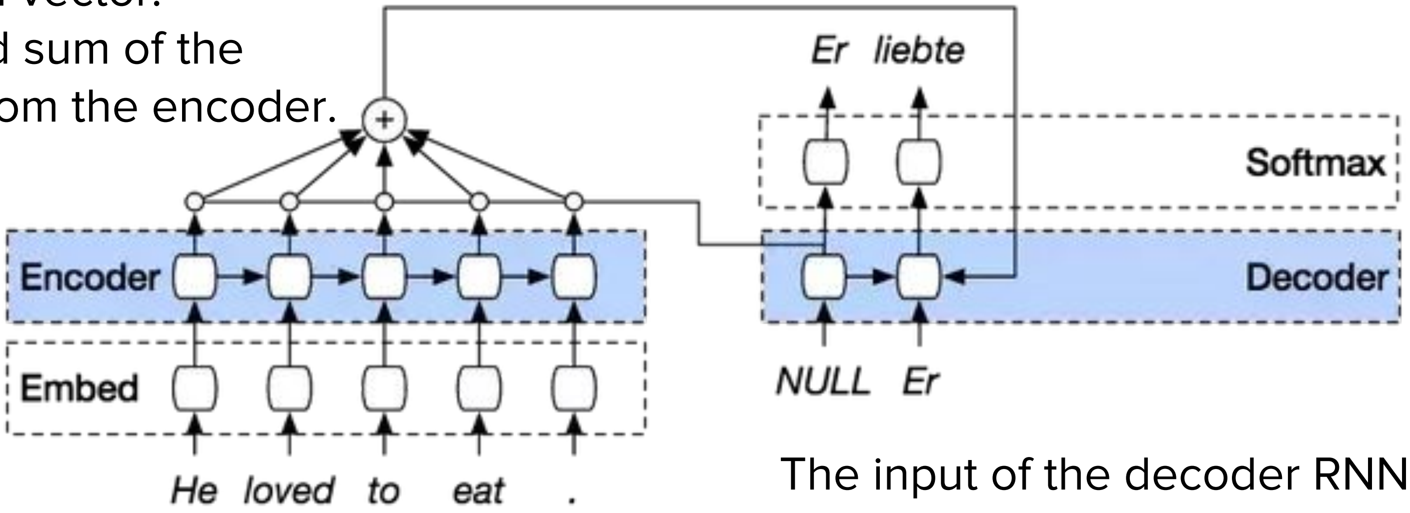
Encoder-decoder structure.



The input of the decoder RNN at is the hidden state of the previous time step and the previous output.

Encoder-decoder with Attention

Attention vector: weighted sum of the output from the encoder.



The input of the decoder RNN at is the hidden state of the previous time step, the previous output and the **attention vector.**

Analysis of Word Embeddings

300 dimension word2vec embeddings trained on the 100B word Google news corpus.

Compared to random initialization with unit variance:

- **Small micro variance**: the variance between dimensions of the same word is small

Initialization	L2 norm	Micro Variance	Cosine Sim.
Random	17.3 ± 0.45	$\boxed{1.00} \pm 0.05$	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	$\boxed{0.02} \pm 0.02$	0.13 ± 0.11

Analysis of Word Embeddings

300 dimension word2vec embeddings trained on the 100B word Google news corpus.

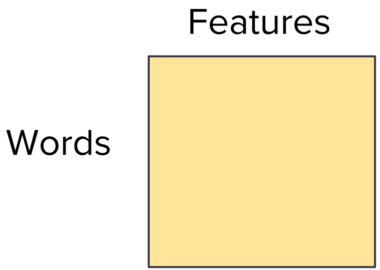
Compared to random initialization with unit variance:

- Small micro variance: the variance between dimensions of the same word is small
- **Large macro variance**: the L2 norm of different words varies largely

Initialization	L2 norm	Micro Variance	Cosine Sim.
Random	$17.3 \pm \boxed{0.45}$	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	$2.04 \pm \boxed{1.08}$	0.02 ± 0.02	0.13 ± 0.11

Embedding Standardization

- Per-example standardization: make variance of each row 1
  - Reduces variance of L2 norm among words
  - Cosine similarity between words is perserved
- Per-feature standardization: make the variance of each column 1
- Per-example normalization: make the L2 norm of each word 1



Initialization	L2 norm	Micro Variance	Cosine Sim.
Random	17.3 ± 0.45	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	0.02 ± 0.02	0.13 ± 0.11
WORD2VEC + ES	17.3 ± 0.05	1.00 ± 0.00	0.13 ± 0.11
WORD2VEC + FS	16.0 ± 8.47	1.09 ± 1.31	0.12 ± 0.10
WORD2VEC + EN	1.00 ± 0.00	0.01 ± 0.00	0.13 ± 0.11

Experiments: Dataset

Dataset contains 8 different domains.

The mapping from canonical utterances to logical forms are given.

The input utterances are collected via crowdsourcing.

Metric	CALENDAR	BLOCKS	HOUSING	RESTAURANTS	PUBLICATIONS	RECIPES	SOCIAL	BASKETBALL
# of example (N)	837	1995	941	1657	801	1080	4419	1952
# of logical form ($\lvert\mathcal{Z}\rvert$, $\lvert\mathcal{C}\rvert$)	196	469	231	339	149	124	624	252
vocab. size ($\lvert\mathcal{V}\rvert$)	228	227	318	342	203	256	533	360
% $\in$ other domains	71.1	61.7	60.7	55.8	65.6	71.9	46.0	45.6
% $\in$ WORD2VEC	91.2	91.6	88.4	88.6	91.1	93.8	86.9	86.9
% $\in$ other domains + WORD2VEC	**93.9**	**93.8**	**90.9**	**90.4**	**95.6**	**97.3**	**89.3**	**89.4**

Baselines

1. (Wang et al) Log-linear model.
2. (Xiao et al) Multi-layer perceptron to encode the unigrams and the bigrams of the input, and then use a RNN to predict the logical form.
3. (Jia and Liang) Seq2Seq model (bi-RNN with attentive decoder) to predict the linearized logical form.
4. (Herzig and Berant) Use all domains to train a single parser with a special encoding to differentiate between domains.

Experiments: Single Domain

Method	Avg. Accuracy
Wang et al.	58.8
Xiao et al.	72.7
Jia and Liang	75.8
Random + I	75.7

Random +I is the most basic model using random initialization of word embeddings.

This model is comparable to previous single domain models.

Experiments: Cross-Domain

Model	Avg Accuracy
Herzig and Berant	79.6
Random	76.9
Word2Vec	74.9
Word2Vec +EN	71.2
Word2Vec +FS	78.9
Word2Vec +ES	80.6

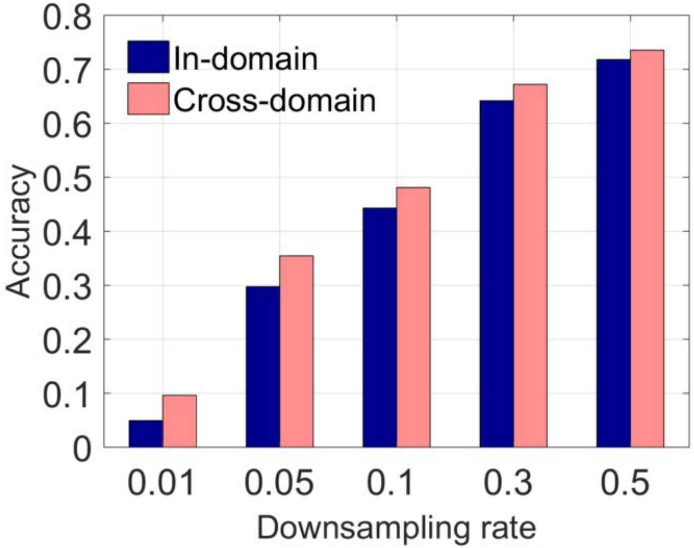
1. Directly using Word2Vec pretrained vectors hurts!
2. Per-example normalization (EN) decreases performance even more.
3. Both per-feature standardization(FS) and per-example standardization(ES) improves performance. Per-example standardization works better.

The perfomance gain is mainly due to word embedding standardization.

Other results

The improvement of cross-domain training is more significant when the target domain data is scarce.

The in-domain training data is downsampled.



Discussion on Standardization/Normalization

> Normalization improves performance in similarity tasks. (Levy et al. 2015)

> A word that is consistently used in a similar context will be represented by a longer vector than a word of the same frequency that is used in different contexts. The L2 norm is a measure of word significance. (Wilson and Schakel 2015)

It is worth trying different normalization schemes for your task!

Conclusion

1. The semantic parsing problem can be decomposed into two steps: first paraphrase the utterance into a canonical form, then translate this canonical form into logical form (idea from Berant and Liang, 2014)
2. Paraphrasing can be learned by a seq2seq model. (We can formulate paraphrasing as translation)
3. Initialization of word embeddings is critical for performance.
4. Out-of-domain data may be useful to improve in-domain performance. (transfer learning philosophy)

References

- Su, Yu and Xifeng Yan. “Cross-domain Semantic Parsing via Paraphrasing.” *EMNLP*(2017).
- Berant, Jonathan and Percy Liang. “Semantic Parsing via Paraphrasing.” *ACL* (2014).
- Wang, Yushi et al. “Building a Semantic Parser Overnight.” *ACL* (2015).
- Herzig, Jonathan and Jonathan Berant. “Neural Semantic Parsing over Multiple Knowledge-bases.” *ACL* (2017).
- Jia, Robin and Percy Liang. “Data Recombination for Neural Semantic Parsing.”ACL (2016)
- Xiao, Chunyang et al. “Sequence-based Structured Prediction for Semantic Parsing.” *ACL* (2016).