Extracting Drug-Drug Interactions with Word and Character-Level Recurrent Neural Networks

Ramakanth Kavuluru, Anthony Rios, and Tung Tran

Division of Biomedical Informatics, Internal Medicine
Department of Computer Science

Fifth IEEE International Conference on
Healthcare Informatics (ICHI 2017)



Outline

1. Motivation
2. Methods
3. Experiments
4. Results

What are Drug-Drug Interactions (DDIs)?

- DDIs represent an important category of adverse drug reactions (ADRs)
- Occurs when effects of one drug is changed by the presence of another drug
- Example: Ephedrine enhances the antinociceptive effect of dexmedetomidine in mice .

DDI Extraction Task

- Given: A sentence annotated with drug mentions
- Objective: Extract triples (Subject,Predicate,Object) where
  - Subject and Object are pharmacological substances
  - Predicate is the class of interaction (more later)

Motivation

- There exists curated databases with DDIs
- However, most up-to-date information about DDIs can still be found in unstructured text
- It is important to extract such interactions from
  - Research articles
  - Warnings in drug labels
  - Observations in clinical notes

Contributions

1) We experiment with RNNs at the character level

- Intuitively good for generalizing to variations of words
- Character embeddings have shown good results in Named Entity Recognition (NER) and language modeling
- Character-level models tend to use *fewer* parameters

Contributions (continued)

2) We study average model behavior with bootstrapped ensembling

- Different parameter initializations of deep neural networks lead to potentially different local minimas during training
- Performance is not deterministic for a fixed training and testing set
- Model averaging is used to arrive at a more consistent model
- Overall, this aspect has not been addressed in prior efforts

The 2013 DDI extraction challenge

- A new dataset for extracting DDIs from free-text narratives
- Consists of Medline abstracts and DrugBank narratives from a total of 792 documents
- Training set has 27,792 candidate pairs
  - 15% are positive for an interaction
- Testing set has 5,716 candidate pairs
  - 17% are positive for an interaction
- Four types of interaction: effect, mechanism, advise, and int

The 2013 DDI extraction challenge (continued)

	Description	Distribution
effect	**a consequence of an interaction is specified** *example: the antihypertensive effect of A may be blunted by the non-steroidal anti-inflammatory drug B*	41%
mechanism	**a pharmacokinetic mechanism is explicity discussed** *example: A decreases the elimination of B causing an increase in overall exposure*	32%
advise	**suggestions regarding handling a drug are made** *example: Patients should be warned of the potential danger of the self-administration of A while under treatment with B*	21%
int	**interaction discussed without any specific additional information** *example: A may interact with B*	6%

Table: Overview of interaction classes

Preprocessing - Entity Binding

- Replace mentions of the drug pair with DRUGA and DRUGB, and replace mentions of all other drugs with DRUGN

- Example: For a candidate pair (benzodiazepines, subutex)

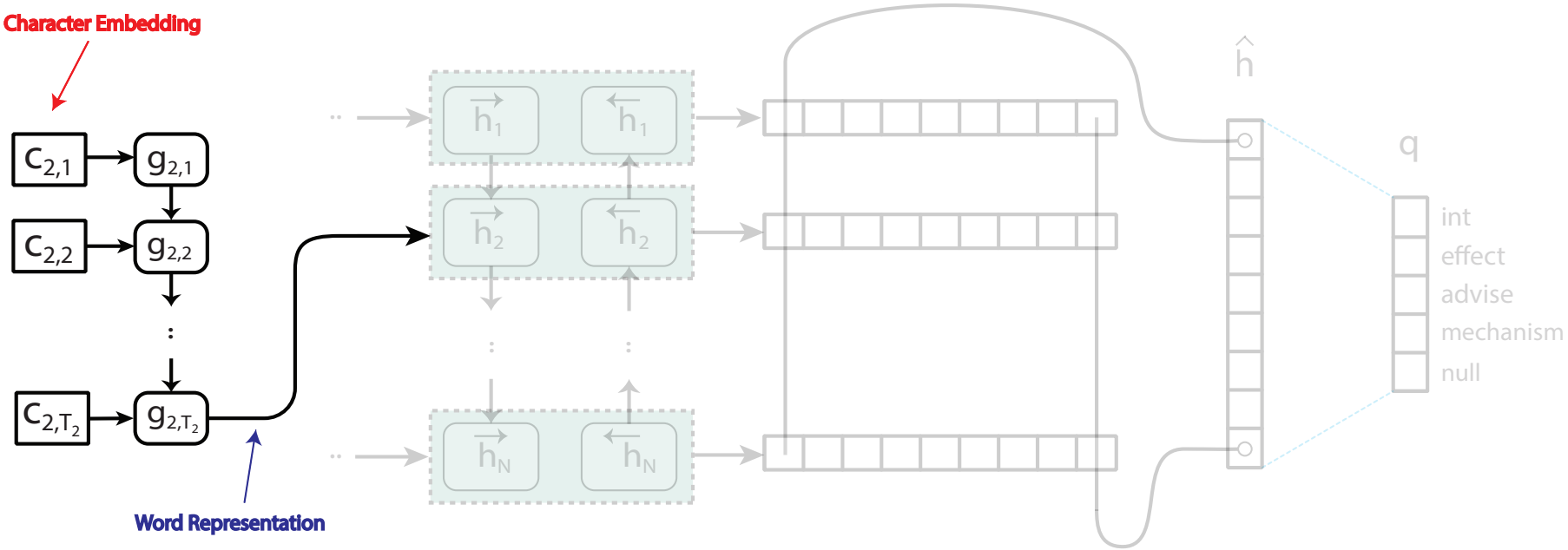
BEFORE

Patients should be warned of the potential danger of the intravenous self-administration of benzodiazepines while under treatment with suboxone or subutex.

AFTER

Patients should be warned of the potential danger of the intravenous self-administration of DRUGA while under treatment with DRUGN or DRUGB.

Network Architecture



1 Character-based LSTM

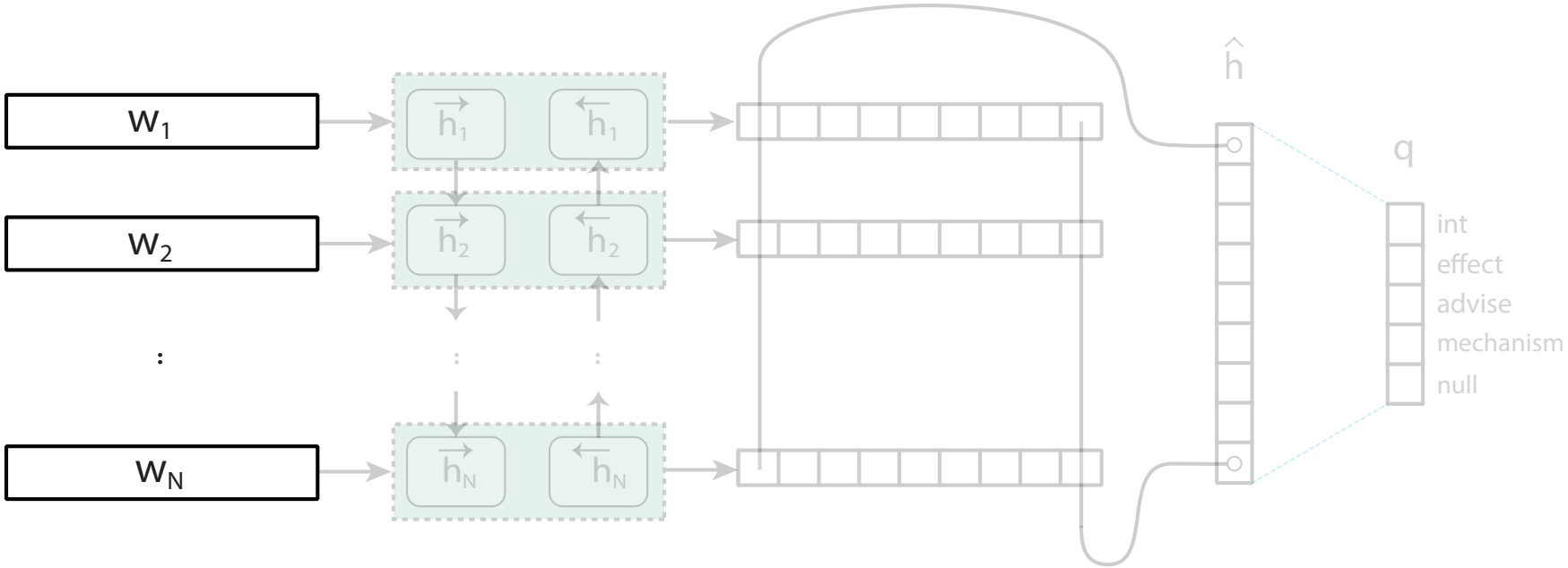
- Forward LSTM over each character embedding of a word
- Output at last timestep forms word representation
- Forms a "hierarchical" composition

OR

Word-based LSTM

- Use pre-trained word embeddings directly

Network Architecture



1 Character-based LSTM

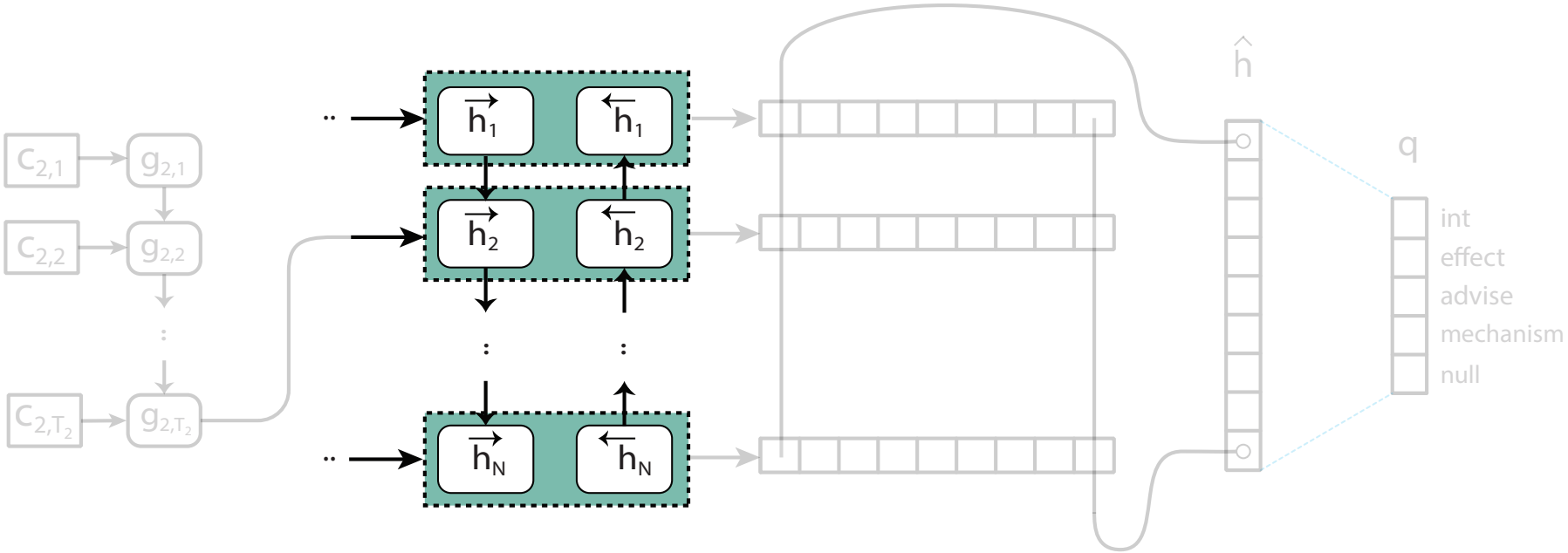
- Forward LSTM over each character embedding of a word
- Output at last timestep forms word representation
- Forms a "hierarchical" composition

OR

Word-based LSTM

- Use pre-trained word embeddings directly

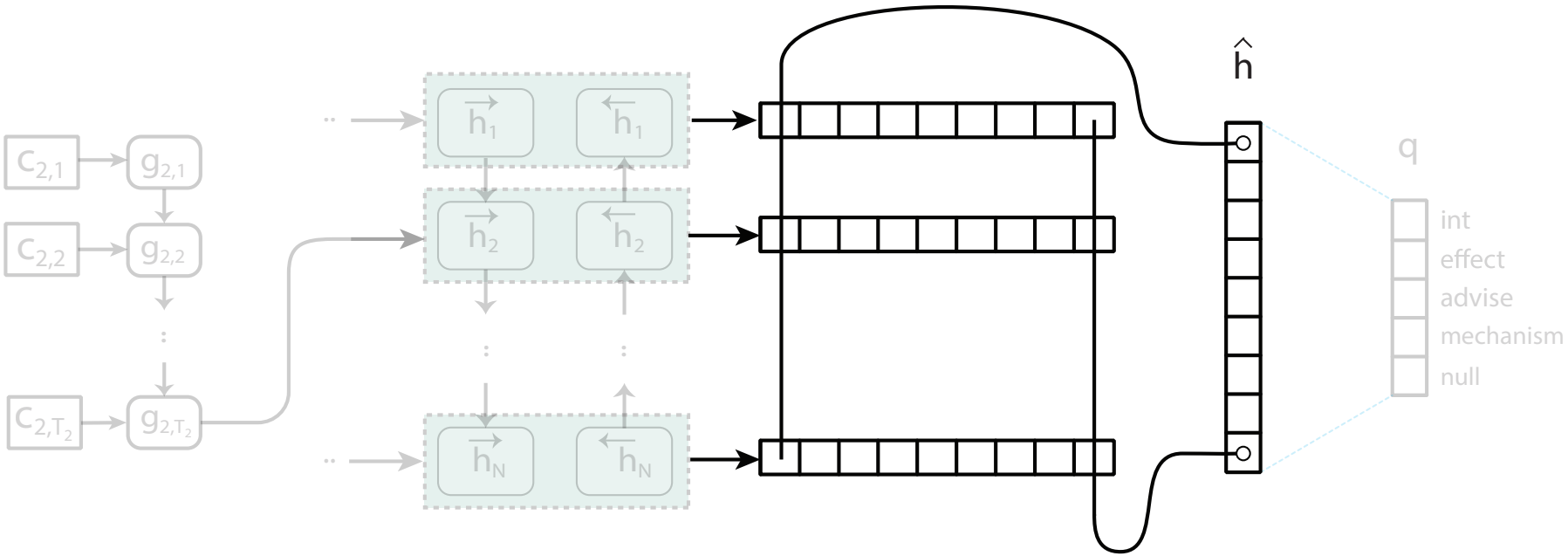
Network Architecture



(2) Contextual Word Representation

- Bidirectional LSTM over each word representation to form "context vectors"
- Each "context vector" represents the entire context centered at some word

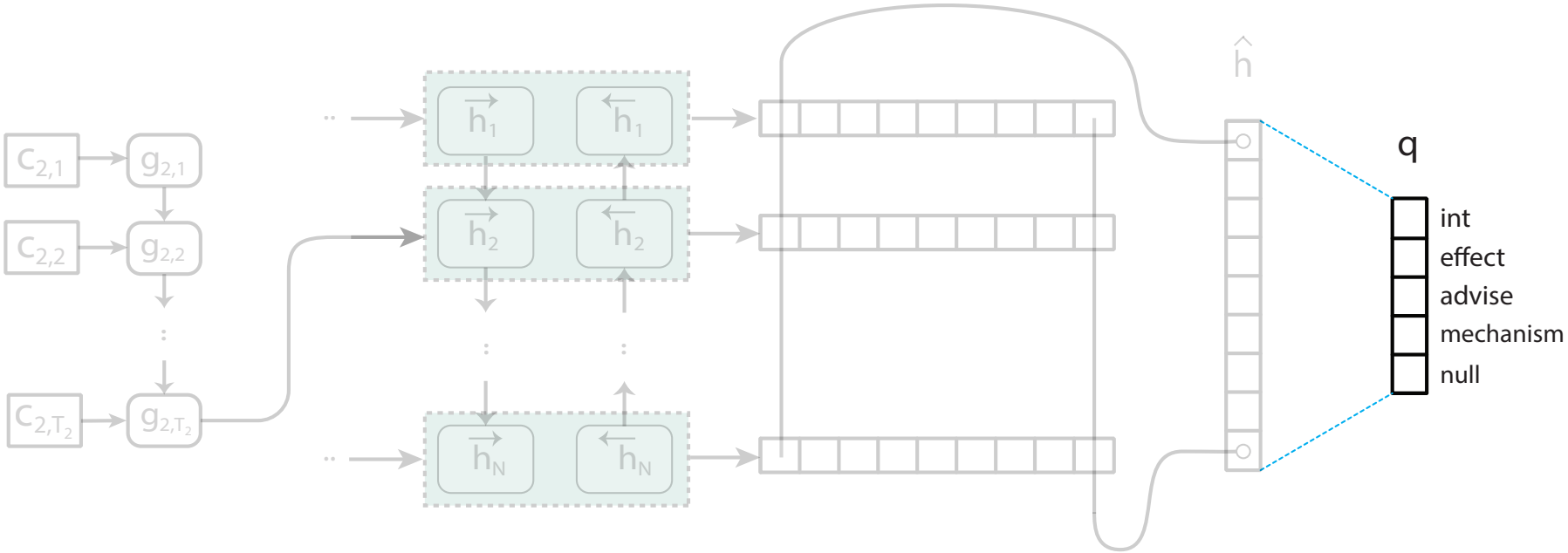
Network Architecture



3 Max-pooling Layer

- Perform a maxpooling operation over each context vector
- Intuitively, this captures only the strongest signals relevant to task

Network Architecture



4 Softmax Output

- Fully connected with one unit per class
- Softmax function results in a categorical distribution
- Trained using cross entropy loss

Position vectors

- We use so-called *position vectors* which has been previously shown to improve performance in relation extraction
- Position vectors represent the relative location of a word with respect to each drug mention
- We concatenate *two* position vectors to each word representation

Experimental Setup

- We train the following models each with a random parameter initialization
  - 20 Word-RNN models
  - 20 Char-RNN models

  (exact hyperparameter settings can be found in full paper)
- We build 10-model ensembles using the following configurations:
  1. Sampling 10 from Word-RNN models
  2. Sampling 10 from Char-RNN models
  3. Sampling 5 each from both models
- For each configuration, we perform the sampling 10,000 times and report mean micro-F1 (MMF)

Negative Filtering

- We additionally run experiments with negative filtering which has been shown to improve performance
- Intuitively, negative filtering is used to remove obvious negative instances with the goal of reducing label imbalance
- In past work, filtering tends to be heavy and obscure
  - Rules may remove up to 50% of the negative test instances
  - Only general guidelines are provided, not an exhaustive list of rules
  - Such heavy filtering may not be generalizable

Negative Filtering (continued)

- We opt for 16 simple rules in our negative filtering step:
  - "DRUGA, DRUGB",
  - "DRUGA, DRUGB, and DRUGN",
  - "DRUGA such as DRUGB",
  - "DRUGA such as DRUGN or DRUGB",
  - "DRUGB such as DRUGN or DRUGA", etc.
- About 18% reduction in negative instances
- An exhaustive list of rules can be found in the full paper

Results

Neg Filter	Ensemble Type	Mean P	Mean R	95% CI of MMF
✗	Char-RNNs	77.56	60.32	67.86 ± 0.00832
	Word-RNNs	74.90	62.46	68.11 ± 0.00768
	Both RNNs	77.78	62.21	69.13 ± 0.00919
✓	Char-RNNs	75.80	61.34	67.80 ± 0.00885
	Word-RNNs	76.02	64.44	69.75 ± 0.01103
	Both RNNs	78.56	63.76	**70.38** ± 0.00963

- Negative filtering generally improves performance

Results

Neg Filter	Ensemble Type	95% CI of MMF	Min F	Max F
✗	Char-RNNs	67.86 ± 0.00832	66.28	69.40
	Word-RNNs	68.11 ± 0.00768	66.67	69.31
	Both RNNs	69.13 ± 0.00919	67.09	70.81
✓	Char-RNNs	67.80 ± 0.00885	66.25	69.45
	Word-RNNs	69.75 ± 0.01103	67.79	71.22
	Both RNNs	**70.38** ± 0.00963	68.28	**72.13**

- With filtering, "Both RNNs" perform better than either individually

Results: Comparison with prior efforts

Team	Precision	Recall	F-score
FBK-irst	64.60	65.60	65.10
WBI	64.20	57.90	60.90
Kim et al.	–	–	67.00
Zheng et al.	–	–	68.40
Liu et al.	75.70	64.60	69.75
Liu et al.	78.24	64.66	70.81
Zhao et al.	72.50	65.10	68.60
Our mean scores ($\neg$F)	77.78	62.21	69.13
Our best ensemble ($\neg$F)	79.34	63.94	70.81
Our mean scores ($+$F)	78.56	63.76	70.38
Our best ensemble ($+$F)	**79.68**	**65.88**	**72.13**

- All prior efforts use some form of (heavy) negative filtering
- Unknown if reporting average or best performance (deep learning)

Results: Comparison with prior efforts

Team	Precision	Recall	F-score
FBK-irst	64.60	65.60	65.10
WBI	64.20	57.90	60.90
Kim et al.	–	–	67.00
Zheng et al.	–	–	68.40
Liu et al.	75.70	64.60	69.75
Liu et al.	78.24	64.66	70.81
Zhao et al.	72.50	65.10	68.60
Our mean scores ($\neg$F)	77.78	62.21	69.13
Our best ensemble ($\neg$F)	79.34	63.94	70.81
Our mean scores ($+$F)	78.56	63.76	70.38
Our best ensemble ($+$F)	**79.68**	**65.88**	**72.13**

- Hence, we report both average and best performance
- Our mean score is on par with the best, even with minimal filtering

Summary

- We evaluated the potential of word and character-level RNNs for DDI extraction.
- We conducted experiments to assess average behavior of these RNNs while comparing them to past efforts.
- We demonstrated that our models are superior or on par with prior results, even when considering average behavior with minimal filtering.

 Thank you 

This effort is supported by the National Library of Medicine, Kentucky Lung Cancer Research Program, and National Center for Advancing Translational Sciences