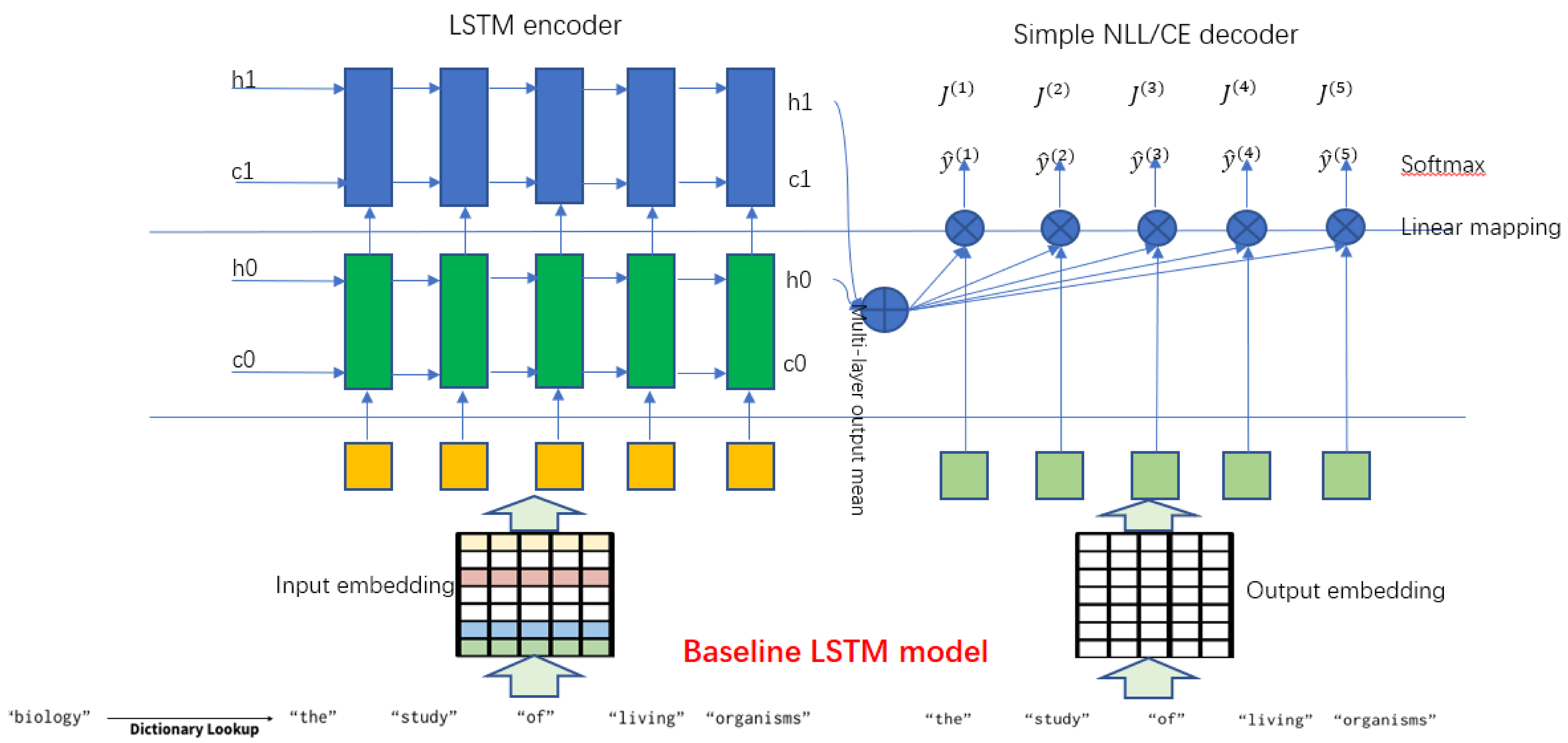
HYBRID DISTRIBUTIONAL AND DEFINITIONAL WORD VECTORS

Haiyuan Mei and Ranjani Iyer
Stanford University, Department of Computer Science

OVERVIEW

- **Motivation**: OOV problem, and exploration of word definitions in downstream NLP tasks.
- **Prior methods:** Def2Vec, on-the-fly embeddings capable of capturing OOV words and limited usage exploration. **?**
- **Approach: HybridVec** Generate word embedding from word definitions, combine it with distributed representations and explore the possibility or improving downstream NLP tasks.
- **Evaluation:** Intrinsic word embedding benchmarks and Extrinsic NMT evaluation, shown to improve translation perplexities and capture complementary aspect of word regarding distributed representation.



MODEL

- **Baseline LSTM:** A two-layer LSTM encoder, Simple linear decoder and NLL loss. Encoder layer hidden output denotes the final definition word vector.
- **Seq2seq:** A two-layer LSTM encoder with dropouts plus a two layer LSTM decoder with attention.
- Input is given as sequence of padded pre-trained word vectors
- The NMT model is a classic Seq2Seq with attention taken directly from **OpenNMT ?**:

TRAINING

- **Dataset: GloVe ?.** All models are trained on pretrained 300d GloVe vectors based on a crawl of 2014 Wikipedia.
  Definitions retrieved from **WordNet?**.
- **HybridVec Implementation:** Pytorch, Adam optimizer, Xavier initialization, hidden size 150, learning rate of 1e-4, batch size 64, 15/20 epochs.
- **Intrinsic evaluation:** Word embedding benchmarks (https://github.com/kudkudak/word-embeddings-benchmarks).
- **NMT Dataset:** OpenNMT-py demo(10k) dataset. Only for comparasion between GloVe and HybridVec.

RESULTS

- Word embeddings benchmarks for GloVe, LSTM Baseline and Seq2seq model. LSTM baseline model is roughly at the level of distributional method; s2s model shows very limited evidence of such capability. (https://github.com/kudkudak/word-embeddings-benchmarks):

	AP	BLESS	Battig	ESSLI_1a	ESSLI_2b	ESSLI_2c	MEN	MTurk	RG65	RW
Glove	0.659204	0.82	0.41120245	0.75	0.825	0.644444	0.737465	0.633182	0.769525	0.366982
baseline glove	0.606965	0.55	0.29325177	0.659091	0.75	0.577778	0.51071	0.422641	0.656402	0.154801
baseline rand	0.549751	0.52	0.27738482	0.613636	0.725	0.533333	0.447908	0.318105	0.644491	0.209295
s2s enc mean	0.221393	0.275	0.1447142	0.522727	0.525	0.4	0.106169	0.137072	0.089082	0.218332

	SimLex99	TR9856	WS353	WS353R	WS353S	Google	MSR	SemEval2012_
Glove	0.3705	0.096717	0.543326	0.477487	0.661995	0.717356	0.61425	0.163963
baseline glove	0.367837	0.122685	0.449105	0.328694	0.542279	0.164603	0.46975	0.112783
baseline rand	0.328812	0.12124	0.35609	0.180284	0.507489	0.170589	0.50625	0.073322
s2s enc mean	-0.01843	0.080803	0.051959	0.025368	0.131758	0.002558	0.008125	0.074485

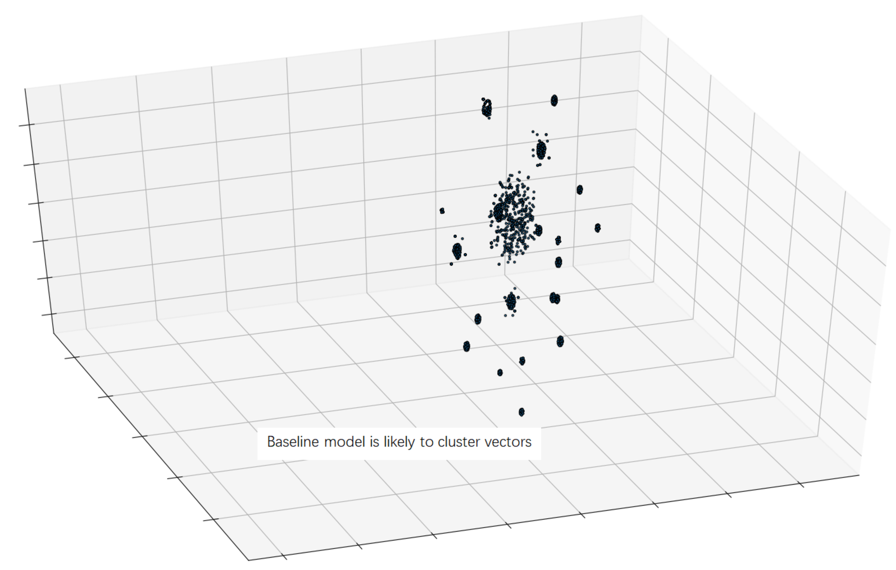
  - GloVe: WEB benchmark for GloVe vectors
  - Baseline glove: WEB benchmark for LSTM baseline model initialized from GloVe
  - Baseline rand: WEB benchmark for LSTM baseline model initialized randomly
  - s2s enc mean: WEB benchmark for Seq2seq model with encoder output mean as the def vec.

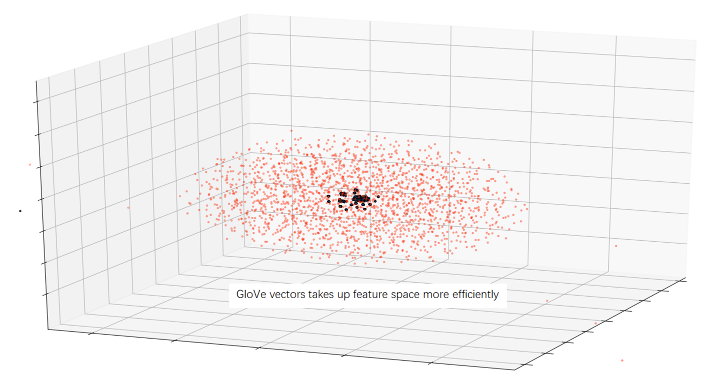
- OpenNMT compare performance improvements using LSTM baseline vector and GloVe:

	No pretrained	LSTM Baseline	GloVe	description
Train PPL	7.47	6.69	4.84	10k nmp demo sentence trained 1 epoch, with/o pretrained word vectors. Glove has most positive impact, LSTM baseline also exhibites positive impact
Train Acc	56.29	58.4	64.32	
BLEU	0.93	1.37	1.99	10K nmt training demo 10 epochs, eval on 3k nmt val sentences,similar result as above perplexity and accuracy

ANALYSIS

- LSTM baseline vectors tend to cluster in feature space. Need to train from a broader source.
- Glove makes use of feature space more efficiently, grasp more sutle meaning of words.





3D tSNE: LSTM Baseline vectors is likely to cluster

tSNE for both: GloVe uses feature space more efficiently

DISCUSSION

- Additional plans for model: greater regularization, inputting multiple definitions, inputting sentence structure, try other embeddings.
- Plan to open source pretrained model and create demo with reverse-dict.xyz

REFERENCES