A Joint Many-Task Model

Growing a Neural Network for Multiple NLP Tasks

Paper: Kazuma Hashimoto && Caiming Xiong && Yoshimasa Tsuruoka && Richard Socher

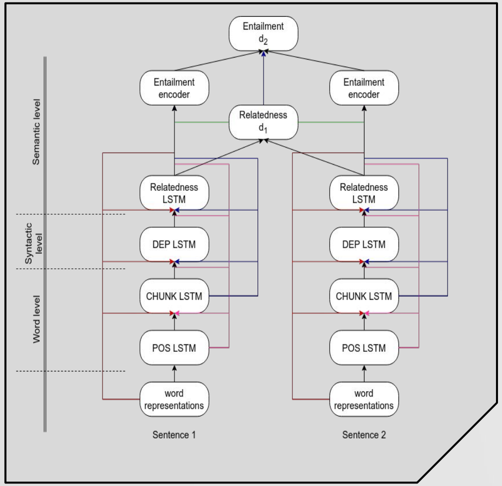
Presentation: Ivan Bardarov && Balint Hompot

Contents

01

Introduction

What are we presenting?



Joint — Trained in an end-to-end fashion

Many-Task — 5 different NLP task hierarchically

Neural Network — All tasks learned by an LSTM

Why?

- Multiple levels of representation to help solve complex tasks
- Hierarchical nature aligns well with human language processing and deep learning models
- Existing systems:
  - ignore linguistic hierarchies
  - are pipelines (not trained end-to-end)

Taxonomy

Network architecture
Hierarchical sharing

Consecutive learning
Task prioritisation

Task weights
No explicit weighing

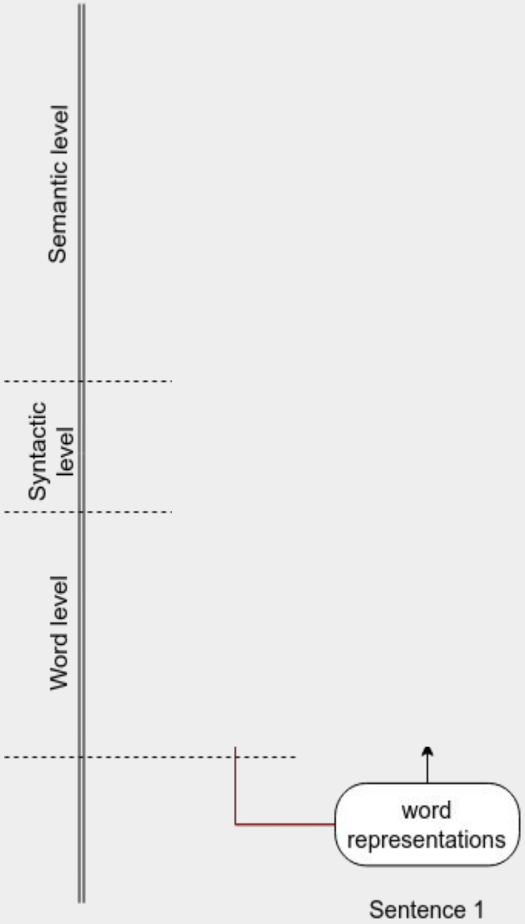
02

Architecture

Architecture

Joint Many-Task (JMT) model

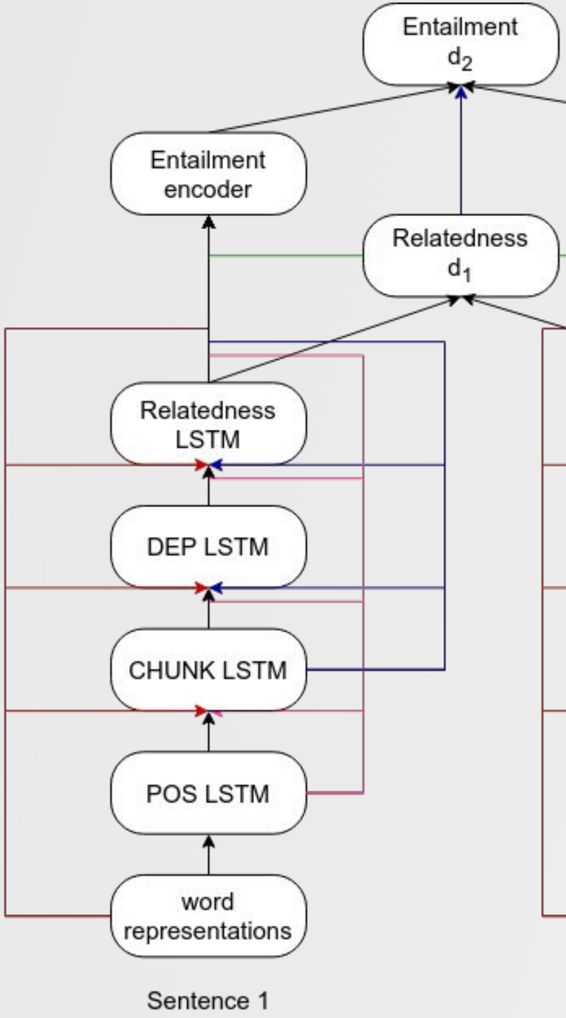
- POS tagging
- Chunking
- Dependency parsing
- Semantic relatedness
- Textual entailment



Modules

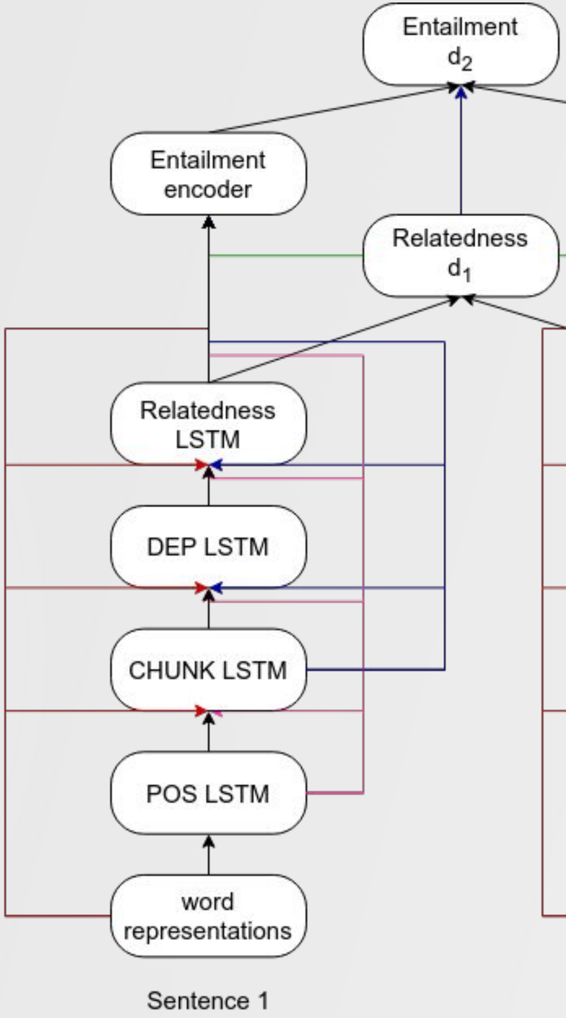
- **POS**
  - Input: embeddings
  - bi-LSTM + ReLU layer + softmax
- **Chunking**
  - Input: embeddings + POS hidden + POS LE
  - bi-LSTM + ReLU layer + softmax

LE = label embedding = $y_t^{(pos)} = \sum_{j=1}^{C} p(y^{(1)} = j | h_t^{(1)}) l(j)$



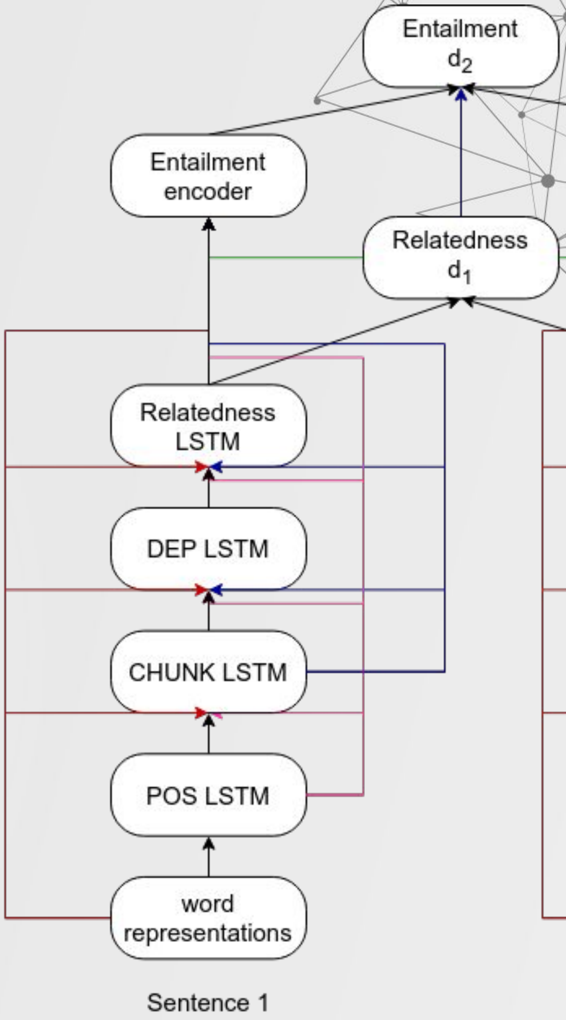
Modules

- **Dependency parsing**
  - Input: embeddings + chunk hidden + POS LE+ chunk LE
  - bi-LSTM +
    - matching function $m(i,j) = h_i^{(3)} \cdot (W_d h_j^{(3)})$
    - ReLU layer + softmax



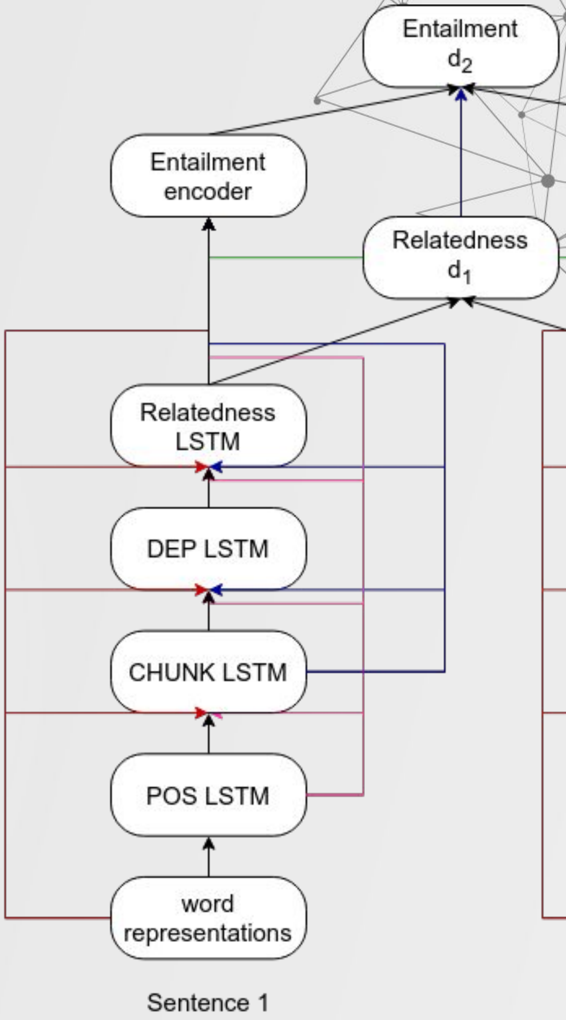
Modules

- **Semantic relatedness**
  - Input: embedding + dep hidden + POS LE + chunk LE
  - bi-LSTM + Max pooling
  - $d_1(s, s') = \left[|h_s^{(4)} - h_{s'}^{(4)}|; h_{s'}^{(4)} \odot h_{s'}^{(4)}\right]$
  - Maxout layer + softmax



Modules

- **Textual entailment**
  - Input: embedding + dep hidden + POS LE + chunk LE + relatedness LE
  - bi-LSTM + Max pooling
  - $d_2(s, s') = \left[h_s^{(5)} - h_{s'}^{(5)}; h_{s'}^{(5)} \odot h_{s'}^{(5)}\right]$
  - 3 Maxout layers + softmax



03

Training

Word representations

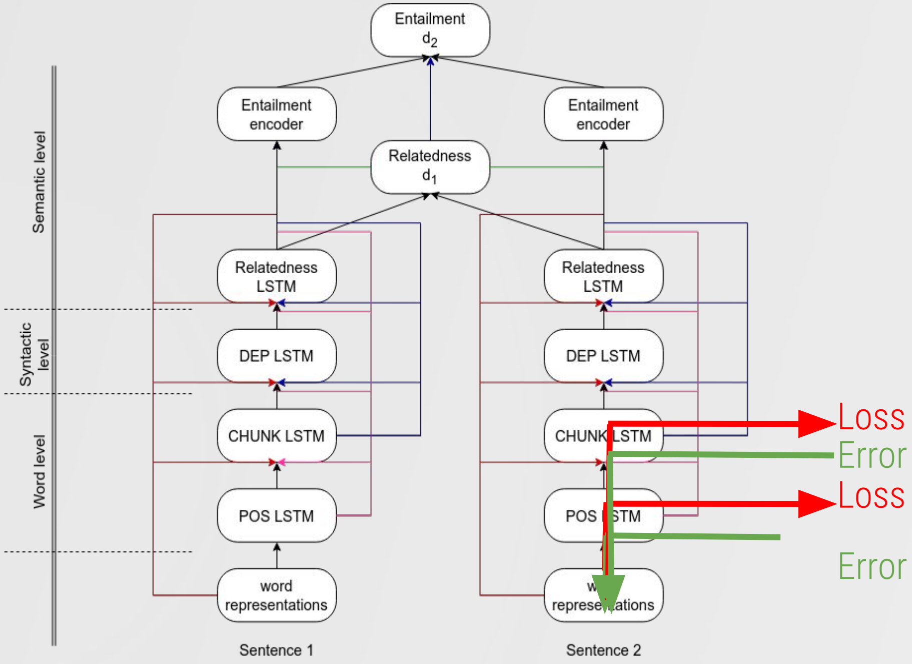
Word embeddings → `Pre-train skip-gram`
Semantics

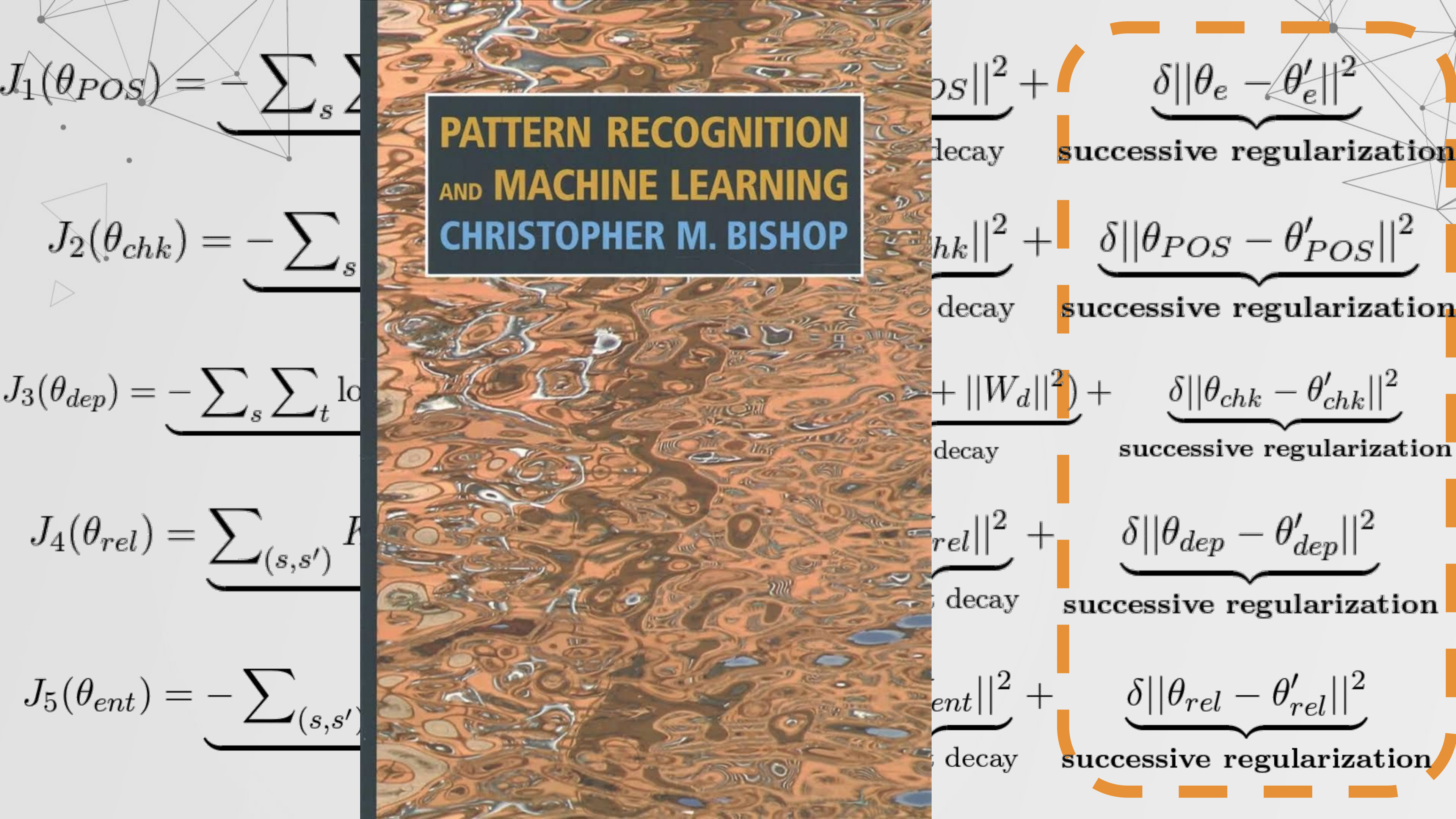
Character n-gram embeddings → `Pre-train skip-gram`
Morphology

Fine tuned

Task order and end-to-end learning

- Consecutive learning:
  - 1 epoch: full dataset on all tasks
  - Bottom to top
- End-to-end:
  - Upper layers dependent on lower
  - Backpropagate




PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP
decay
successive regularization
decay
successive regularization
decay
successive regularization
decay
successive regularization
decay
successive regularization

04

Experiments

Datasets

	Metric
POS: Wall Street Journal (WSJ)	Accuracy
Chunking: WSJ	F1
Dependency parsing: *converted* WSJ	UAS and LAS
Semantic relatedness: SICK	MSE
Text entailment: SICK	Accuracy

Results

		Single	JMT_{all}	JMT_{AB}	JMT_{ABC}	JMT_{DE}	JMT_{CD}	JMT_{CE}
A ↑	POS	97.45	97.55	97.52	97.54	n/a	n/a	n/a
B ↑	Chunking	95.02	n/a	95.77	n/a	n/a	n/a	n/a
C ↑	Dependency UAS	93.35	94.67	n/a	94.71	n/a	93.53	93.57
	Dependency LAS	91.42	92.90	n/a	92.92	n/a	91.62	91.69
D ↓	Relatedness	0.247	0.233	n/a	n/a	0.238	0.251	n/a
E ↑	Entailment	81.8	86.2	n/a	n/a	86.8	n/a	82.4

Method	F1 ↑
JMT_{AB}	**95.77**
Single	95.02
Søgaard and Goldberg (2016)	95.56
Suzuki and Isozaki (2008)	95.15
Collobert et al. (2011)	94.32
Kudo and Matsumoto (2001)	93.91
Tsuruoka et al. (2011)	93.81

Chunking

POS tagging

Method	Acc. ↑
JMT_{all}	97.55
Ling et al. (2015)	**97.78**
Kumar et al. (2016)	97.56
Ma and Hovy (2016)	97.55
Søgaard (2011)	97.50
Collobert et al. (2011)	97.29
Tsuruoka et al. (2011)	97.28
Toutanova et al. (2003)	97.27

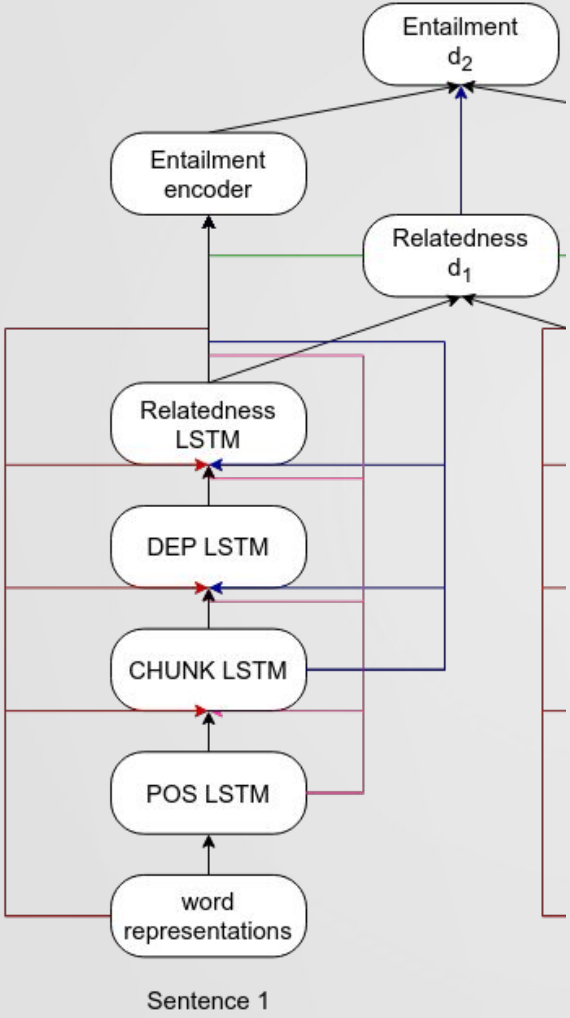
Dependency parsing

Method	UAS ↑	LAS ↑
JMT_{all}	94.67	92.90
Single	93.35	91.42
Dozat and Manning (2017)	**95.74**	**94.08**
Andor et al. (2016)	94.61	92.79
Alberti et al. (2015)	94.23	92.36
Zhang et al. (2017)	94.10	91.90
Weiss et al. (2015)	93.99	92.05
Dyer et al. (2015)	93.10	90.90
Bohnet (2010)	92.88	90.71


WENGER

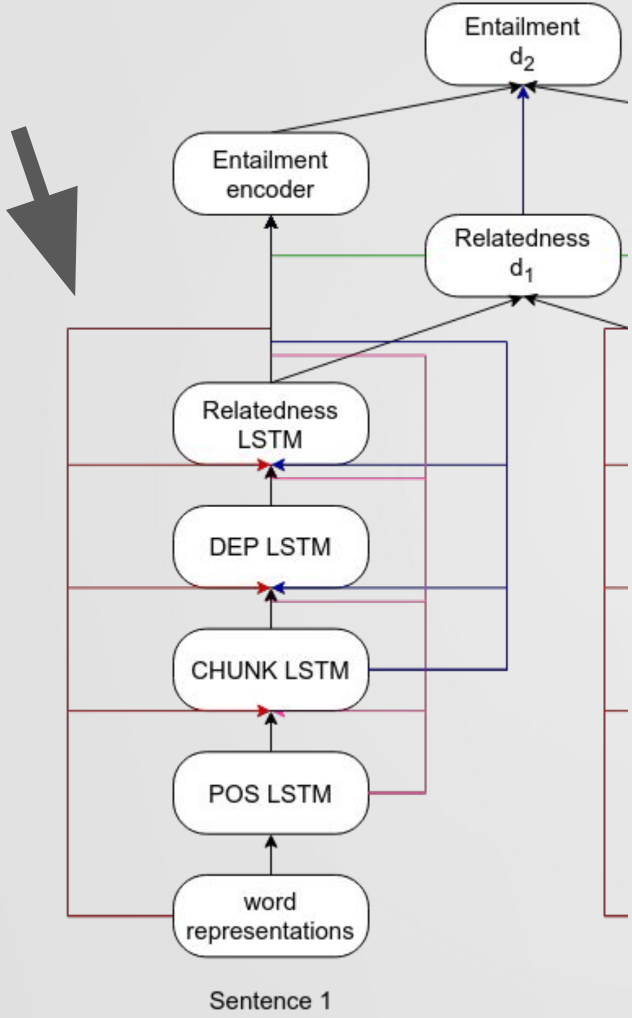
05

Model Analysis



Depth

	Single	Single+
POS	97.52	
Chunking	95.65	96.08
Dependency UAS	93.38	93.88
Dependency LAS	91.37	91.83
Relatedness	0.239	0.665
Entailment	83.8	66.4



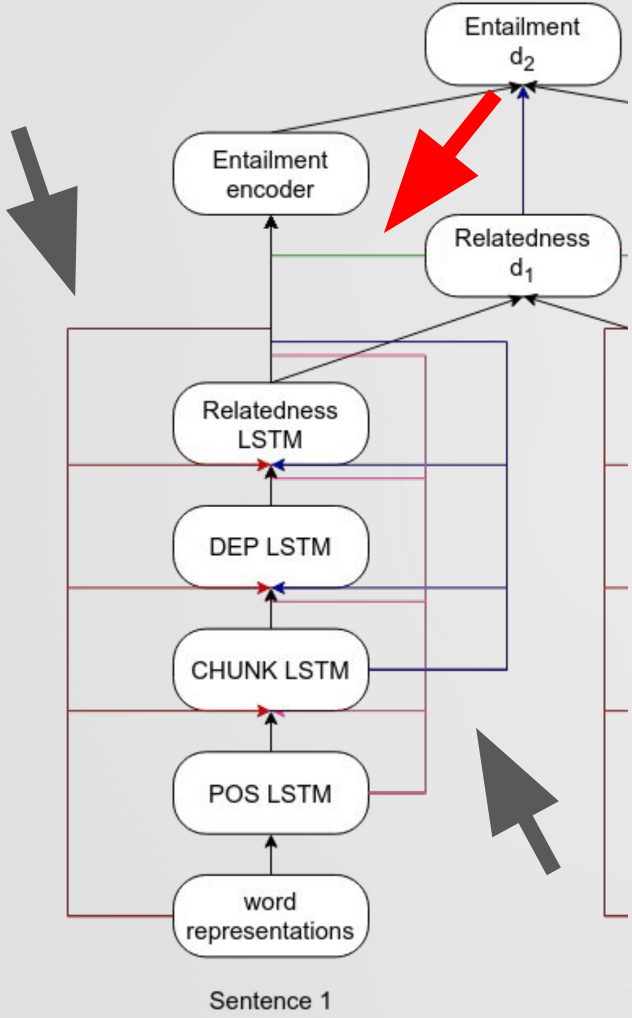
Shortcut connections & Label Encoding

	JMT_{all}	w/o SC
POS	97.88	97.79
Chunking	97.59	97.08
Dependency UAS	94.51	94.52
Dependency LAS	92.60	92.62
Relatedness	0.236	0.698
Entailment	84.6	75.0

Sample: "Standing"

	Word and char	Only word	
Embedding	leaning kneeling saluting clinging railing	stood stands sit pillar cross-legged	**Semantics**
POS	warning waxing dunking proving tipping	ladder rc6280 bethle warning f-a-18	**Nouns**
Chunking	applauding disdaining pickin readjusting reclaiming	fight favor pick rejoin answer	**Verbs**

Dependency	guaranteeing resting grounding hanging hugging	patiently hugging anxiously resting disappointment	**Adverbs + Nouns (Dep on verbs)**
Relatedness	stood stands unchallenged notwithstanding judging	stood unchallenged stands beside exists	**Semantics**
Entailment	nudging skirting straddling contesting footing	beside stands pillar swung ovation	**Semantics**

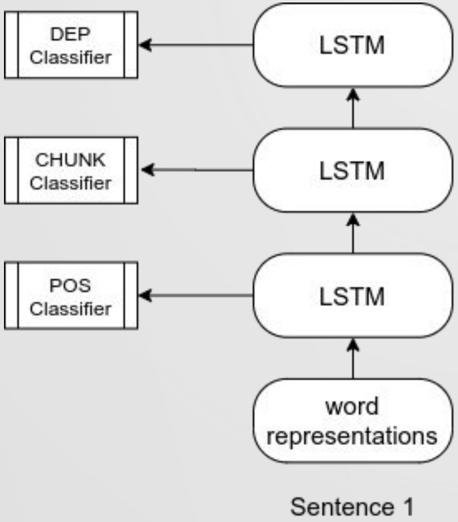


Shortcut connections & Label Encoding

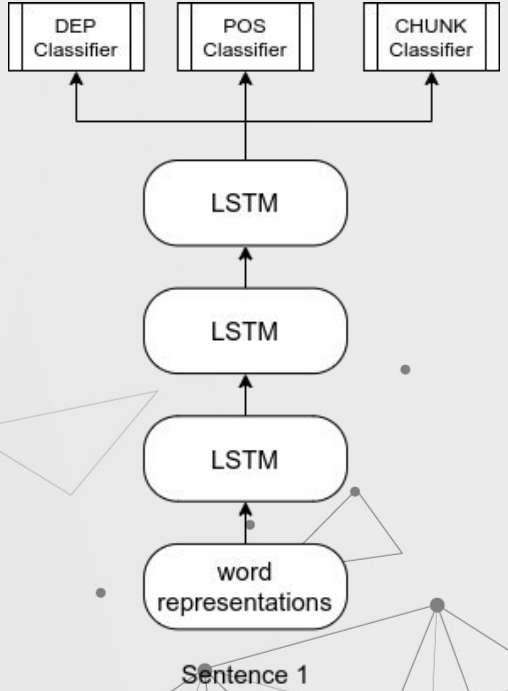
	JMT_{all}	w/o SC	w/o LE	w/o SC&LE
POS	97.88	97.79	97.85	97.87
Chunking	97.59	97.08	97.40	97.33
Dependency UAS	94.51	94.52	94.09	94.04
Dependency LAS	92.60	92.62	92.14	92.03
Relatedness	0.236	0.698	0.261	0.765
Entailment	84.6	75.0	81.6	71.2

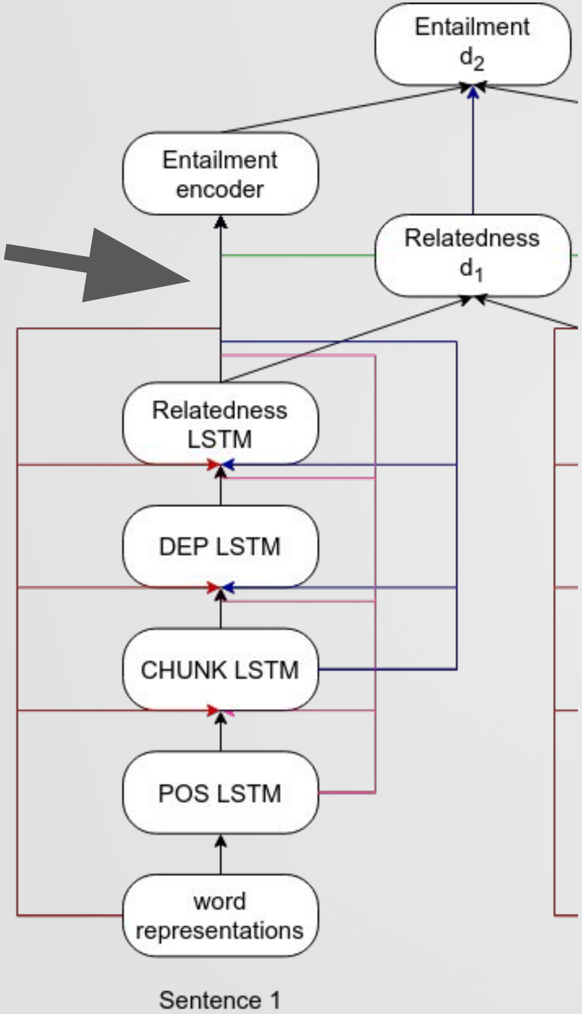
Different layers

	JMT_{ABC}	w/o SC&LE	All-3
POS	97.90	97.87	97.62
Chunking	97.80	97.41	96.52
Dependency UAS	94.52	94.13	93.59
Dependency LAS	92.61	92.16	91.47



Battle





Successive regularization & Vertical connections

	JMT_{all}	w/o SR	w/o VC
POS	97.88	97.85	97.82
Chunking	97.59	97.13	97.45
Dependency UAS	94.51	94.46	94.38
Dependency LAS	92.60	92.57	92.48
Relatedness	0.236	0.239	0.241
Entailment	84.6	84.2	84.8

06

Conclusion and Discussion

Conclusion

- Hierarchical model that improves over hard-parameter sharing ones
- Low-level tasks improve high-level ones and vice versa
- Shortcut connections are crucial

Authors' discussion

Training strategy

- Not obvious when to stop
- Dependency accuracy maximized
- Same number of epochs for all

- Entity detection and relation extraction
- Multiple domains

More tasks

Learn low-level features with a high-level task

- Existing work on learning task oriented latent graph structures of sentences using machine translation

Paper opinion

Positives

- Very well-structured
- Close SOTA on all tasks in the joint mode
- Extensive experimenting and ablation

Room for improvement

- Lacking motivation behind choices
    - Maxout layers

Opinion & Future work

Hierarchy engineering

BERT Rediscovers the Classical NLP Pipeline

Attention for the LSTMs

Connect dependency layer

Character level encoders

THANKS

Does anyone have any questions?

Modelling the interplay of metaphor and emotion through multitask learning

by Verna Dankers, Marek Rei, Martha Lewis, Ekaterina Shutova

Contents

1. Motivation behind the Research
2. Main Contributions
3. Model Architectures & Methodology
4. Experiments
5. Results
6. Discussion & Evaluation of the paper
7. Questions

Metaphors

Definition: “A metaphor is a figure of speech that, for rhetorical effect, directly refers to one thing by mentioning another.“ [Wikipedia]

Often used to express **emotions** in an **abstract** way.

“My mind is seething and boiling”

Your brain does not have a high temperature in a **literal** sense (source)

But you are so angry that it **feels** like your brain is overheating (target)

Metaphors

Humans can even infer the meaning of a metaphor they don’t know due to their capability to emotionally relate

Motivation behind the Research

- Metaphor detection and emotion regression are rather hard NLP tasks
- Evidence from other disciplines (linguistics, cognitive psychology and neuroscience) that metaphors are highly connected to emotions (metaphors are more emotionally evocative)

→ Research Question: Do the two tasks share similar semantic concepts and can they profit from each other in a MTL approach?

Main Contributions

Previous work:

- Mostly **separate** approaches to emotion regression and metaphor detection
- Already tried to incorporate emotion information into metaphor identification

What's new?

- **Joint** MTL approach training for both tasks **at the same time**
- Advances state of the art in **both** tasks

The two Tasks

Metaphor identification:

- sequence labeling task (word-level classification: metaphorical or literal)
- metaphoricity score (sentence-level)

Emotion prediction:

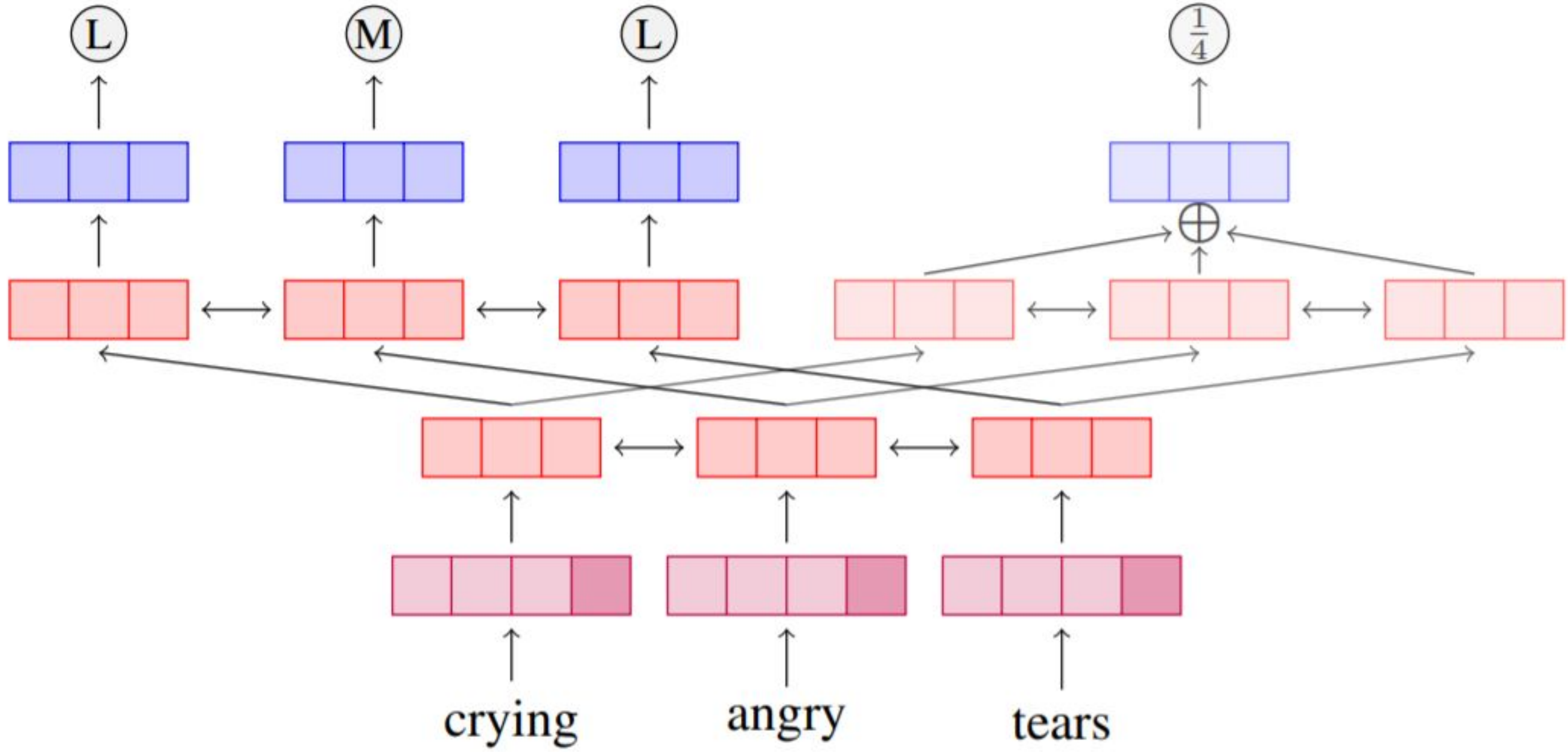
- Sentence-level regression
- Three emotion dimensions:
  Valence (polarity), **A**rousal (strength), **D**ominance (control))

Model Architectures (Joint MTL)

Input: Concatenated GloVe and ELMo word embeddings

1. **Hard parameter sharing**:

- Two shared Bi-LSTM layers for mutual **general** feature extraction
- One task **specific** Bi-LSTM layer (for each of the two tasks)
- Fully-connected layers for classification/regression
- Task specific word-level attention mechanism for sentence-level regression



(a) Hard parameter sharing

Model Architectures (Joint MTL)

- assess effect of MTL independent of model architecture
  → fine-tuned BERT model for comparison

- all transformer layers fixed (hard parameter sharing) except the last layer (task-specific)

Model Architectures (Joint MTL)

2. **Soft parameter-sharing**:
Two separate networks for each task connected to share information

a) Cross-stitching model:

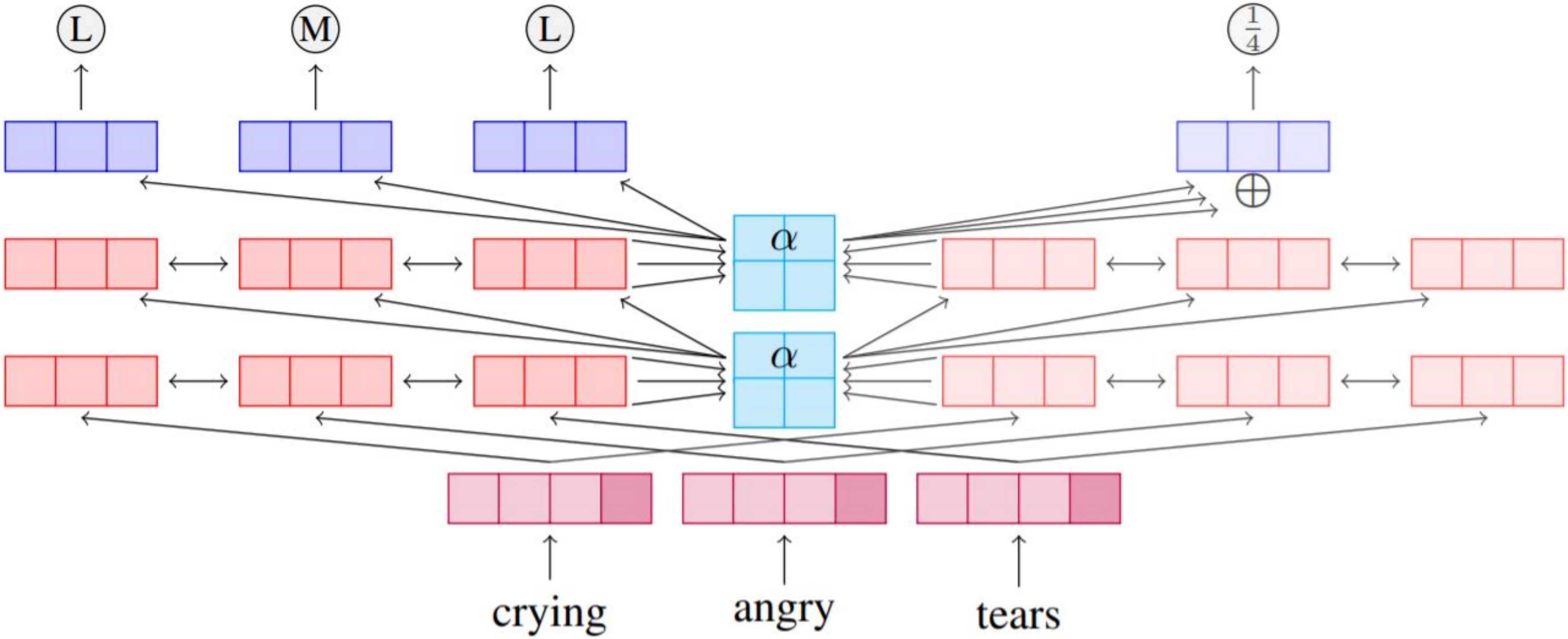
- Three Bi-LSTM layers for each of the two tasks
- Four alpha parameters per layer control information transfer between the two networks

$$\widetilde{\mathbf{h}}_A = \alpha_{AA}\mathbf{h}_A + \alpha_{BA}\mathbf{h}_B$$

From net B to net A

$$\widetilde{\mathbf{h}}_B = \alpha_{BB}\mathbf{h}_B + \alpha_{AB}\mathbf{h}_A$$

From net A to net B



(b) Cross-stitch network

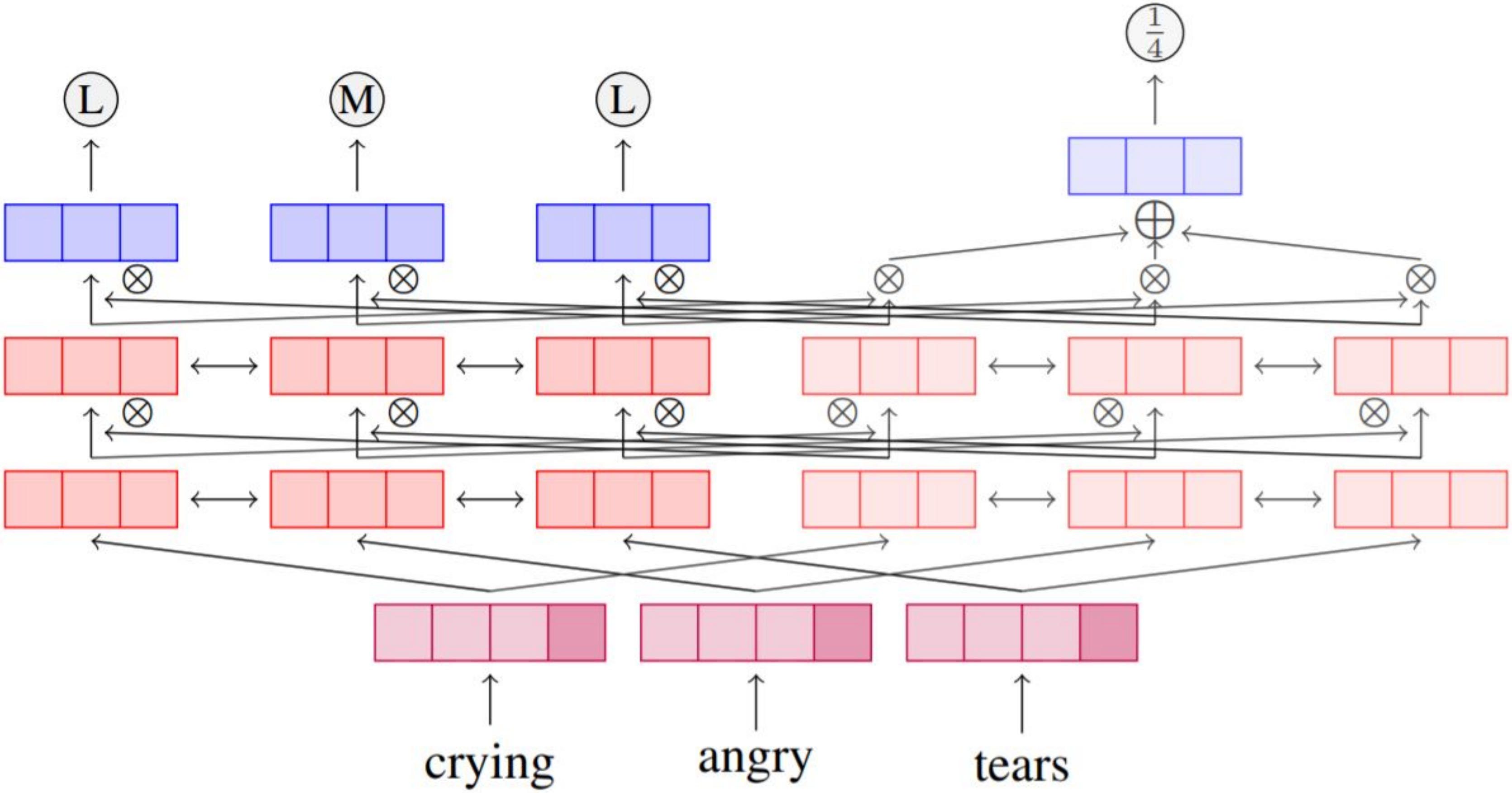
Model Architectures (Joint MTL)

2. Soft parameter-sharing:

b) Gated network:

- similar to the cross-stitch architecture
- BUT replace static globally shared alpha parameters by dynamic gates

$$\mathbf{g}_A = \sigma(\mathbf{W}_A[\mathbf{h}_A; \mathbf{h}_B] + \mathbf{b}_A)$$
$$\widetilde{\mathbf{h}}_A = (1 - \mathbf{g}_A) \odot \mathbf{h}_A + \mathbf{g}_A \odot \mathbf{h}_B$$
$$\mathbf{g}_B = \sigma(\mathbf{W}_B[\mathbf{h}_A; \mathbf{h}_B] + \mathbf{b}_B)$$
$$\widetilde{\mathbf{h}}_B = (1 - \mathbf{g}_B) \odot \mathbf{h}_B + \mathbf{g}_B \odot \mathbf{h}_A$$



(c) Gated network

Contents

Experiments - Datasets

3 Datasets:

1. **VUA metaphor corpus**: >10,000 english sentences from 4 genres (news, conversation academic writing and fiction); binary labels on word level (L, M)

2. **LCC metaphor corpus**: ~9,000 samples from english portion of sentences; sentence-level regression with metaphoricity score

3. **EmoBank corpus**: 10,000 english sentences from many different genres annotated in the VAD emotion dimensions for sentence-level regression.

Experiments - EmoBank Examples

Sentence	Val.	Arous.	Dom.
"Tell her I love her."	.94	.88	.83
Tell me, or I'll kill –	.35	.69	.83
What did you say?	.50	.54	.50
This is torture.	.14	.72	.27

Table 1: EmoBank examples with normalised scores, illustrating the differences among the dimensions.

Experiments - Procedure

- Train each architecture in a STL and MTL setup
- Train emotion dimensions separately
- randomly select one of the two tasks for MTL
- auxiliary task is downscaled to constitute 10% of the loss of the main task
- BCE loss for sequence labeling MSE for regression tasks

Results - Metaphor

- with Dominance MTL consistently outperforms the STL setup
- BERT model gives most improvement
- slight advantage of gated network
- advances state-of-the-art

Approach	Metaphor Task Word (F_1)	Sent. (r)
Gao et al. (2018)	.726	-
LSTM (single task)	.737	.544
Hard Sharing		
+ Valence	.740	**.559**
+ Arousal	.740	**.558**
+ Dominance	**.743**	**.560**
Cross-Stitch Network		
+ Valence	.741	**.556**
+ Arousal	.740	**.558**
+ Dominance	**.743**	**.563**
Gated Network		
+ Valence	**.742**	**.561**
+ Arousal	.741	**.558**
+ Dominance	**.745**	**.560**
BERT (single task)	.763	.604
Hard Sharing		
+ Valence	**.769**	**.614**
+ Arousal	.765	.610
+ Dominance	**.768**	**.614**

Table 2: System performance for the word- and sentence-level metaphor tasks using the F_1-score and Pearson's r respectively. Statistically significant ($p < 0.05$) differences to the single task models are shown in boldface.

Results - Emotion

- for Dominance and Valence MTL consistently outperforms the STL setup
- BERT model gives most improvement
- no big difference between different parameter sharing methods
- advances state-of-the-art

Approach	Emotion Task		
	Val.	Arous.	Dom.
Akhtar et al. (2018)	.616	.355	.237
+ Val., Arous., Dom.	.635	.375	.277
Wu et al. (2019)†	.620	.508	.333
LSTM (single task)	.728	.557	.373
Hard Sharing			
+ Metaphor (Token)	**.734**	**.564**	**.384**
+ Metaphor (Sent.)	**.734**	.558	**.388**
Cross-Stitch Network			
+ Metaphor (Token)	**.737**	**.564**	**.388**
+ Metaphor (Sent.)	**.735**	.558	**.384**
Gated Network			
+ Metaphor (Token)	**.738**	**.563**	**.389**
+ Metaphor (Sent.)	**.735**	.560	**.384**
BERT (single task)	.771	.565	.403
Hard Sharing			
+ Metaphor (Token)	**.779**	**.572**	**.420**
+ Metaphor (Sent.)	**.778**	**.570**	**.417**

Table 3: System performance for emotion regression tasks according to Pearson's r. Statistically significant ($p < 0.05$) differences to the single task model are shown in boldface. †Used 40% of the gold labels.

Discussion

- Dominance dimension most important for metaphors although often ignored by a lot of previous work while Arousal not so important

- Transformer model outperforms recurrent approaches
  → contextual information seems to be important

- Improvement due to MTL setup rather than specific architecture

- Also a **lot** of improvement in emotion regression
  → both way synergy while previous work mostly considered emotion to help metaphor detection

Discussion - Gating Mechanisms

- Gating more open in lower layers while almost no information transfer in the top layers
- Fulfills intuition from general to specific like in hard parameter sharing
- Probably that is why there is little difference between the parameter sharing methods

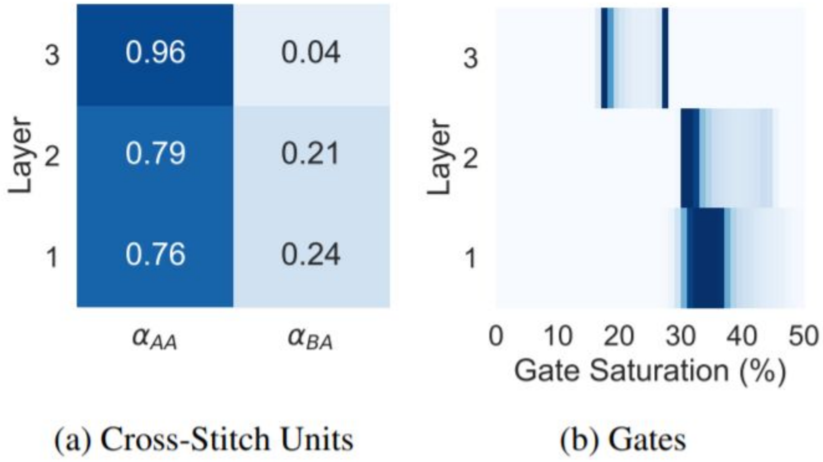


Figure 2: Illustration of the information flow in between the Bi-LSTM layers, for the dominance regression (B) and metaphor identification (A) tasks. Gate saturation % is calculated by averaging across the hidden dimensionality for every word in the test set.

Discussion - Success and Failure

- Improvement mostly from correcting literal STL predictions to metaphorical
- Different key words for the emotion dimensions
- Metaphor detection benefits from the emotion in valence/arousal words and the emotional context of dominance words
- Also some new failure cases introduced by making non-emotional metaphors literal

Conclusion

First MTL approach to jointly model metaphor detection and emotion prediction in text

experiment with various MTL schemes

Metaphor detection

enhances

Emotion prediction
(especially **Dominance**)

Implication: metaphor might be good MTL support for sentiment analysis

Evaluation of the Paper (our opinion)

Pros:

+ Well structured, nice figures, well explained, easy to read
+ detailed information about data pre-processing, hyperparameters, etc.
+ Impressive results: beat state of the art in both tasks

Cons:

- would have been more consistent to also combine the other MTL architectures with a BERT version
- it isn't addressed why the STL setups are already better than previous SotA

Thank you for your Attention!

Questions?

References

Verna Dankers, Marek Rei, Martha Lewis and Ekaterina Shutova (2019). Modelling the interplay of metaphor and emotion through multitask learning. In Proceedings of EMNLP 2019.

Image References

Yellow from the egg:
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.amazon.de%2FShirtcity-English-Not-Yellow-T-Shirt%2Fdp%2FB00TGF2742&psig=AOvVaw24G_H_nDZdwcibUA7is8D4&ust=1587548194870000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCMitplSc-egCFQAAAAAdAAAAABAD

BERT:
https://miro.medium.com/max/1190/1*nWbf_KWFnSNfbeMZSiMrJw.png