Capsule Networks for NLP

Will Merrill

Advanced NLP

10/25/18

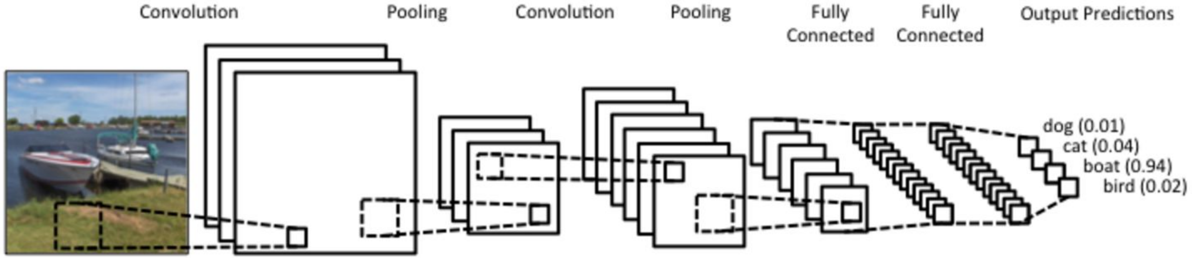
Capsule Networks: A Better ConvNet

- Architecture proposed by Hinton as a replacement for ConvNets in computer vision
- Several recent papers applying them to NLP:
  - Zhao et al., 2018
  - Srivastava et al., 2018
  - Xia et al. 2018
- Goals:
  - Understand the architecture
  - Go through recent papers

What’s Wrong with ConvNets?

Convolutional Neural Networks

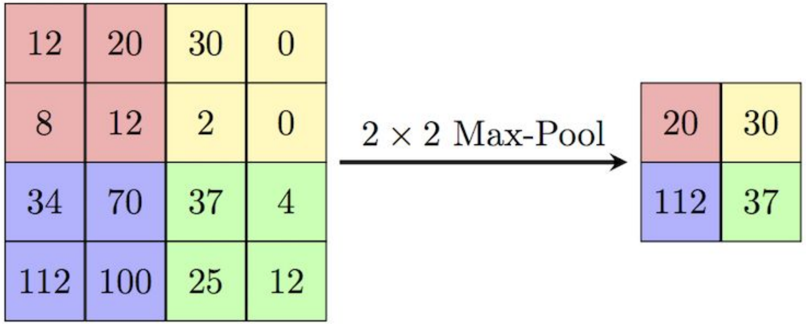
- Cascade of convolutional layers and max-pooling layers
- Convolutional layer:
  - Slide window over image and apply filter



https://towardsdatascience.com/build-your-own-convolution-neural-network-in-5-mins-4217c2cf964f

Max-Pooling

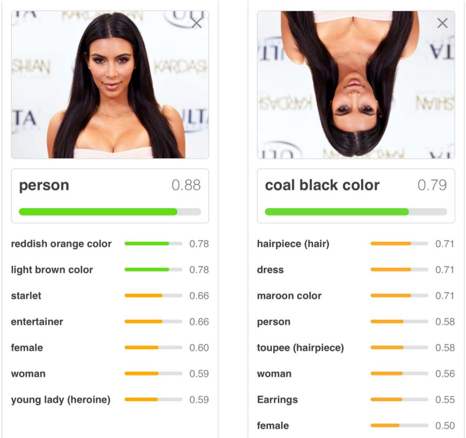
- ConvNets use max-pooling to move from low-level representations to high-level representations



https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

Problem #1: Transformational Invariance

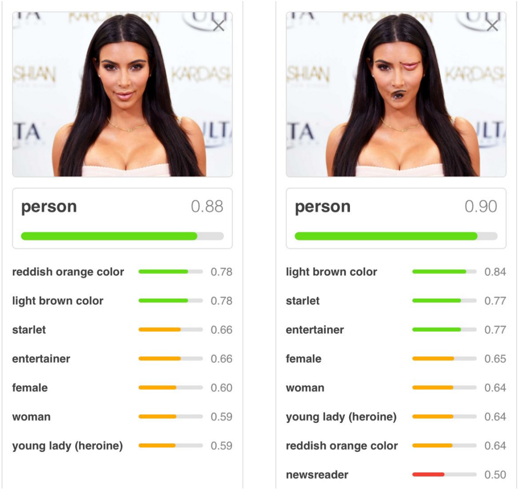
- We would like networks to recognize transformations of the same image
- Requires huge datasets of transformed images to learn transformations of high-level features



https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc

Problem #2: Feature Agreement

- Max-pooling in images loses information about relative position
- More abstractly, lower level features do not need to “agree”



https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc

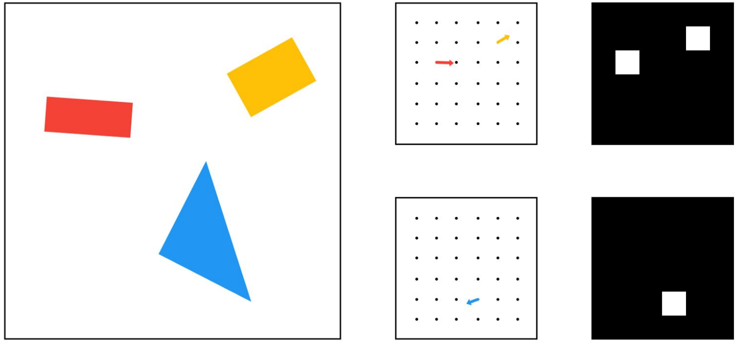
Capsule Network Architecture

Motivation

- We can solve problems #1 and #2 by attaching “instantiation parameters” to each filter
    - ConvNet: Is there a house here?
    - CapsNet: Is there a house with width *w* and rotation *r* here?
- Each filter at each position has a vector value instead of a scalar
- This vector is called a capsule

Capsules

- The value of capsule i at some position is a vector $\mathbf{u}_i$
- $|\mathbf{u}_i| \in (0, 1)$ gives the probability of existence of feature i
- Direction of $\mathbf{u}_i$ encodes the instantiation parameters of feature i



https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc

Capsules (Continued)

https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc

Capsule Squashing Function

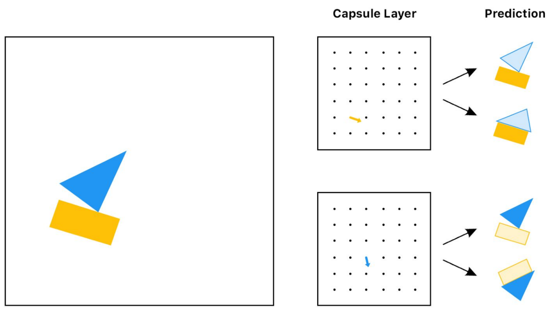
- New squashing function which which puts magnitude of vector into (0, 1)
- Referred to in literature as *g*(..) or `squash`(..)
- Will be useful later on

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

Sabour et al., 2017

Routing by Agreement

- Capture child-parent relationships
- Combine features into higher-level ones only if the lower-level features “agree” locally
- Is this picture a house or a sailboat?



https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc

Routing: Vote Vectors

- Learned transformation for what information should be “passed up” to the next layer
- Models what information is relevant for abstraction/agreement
- $\hat{\mathbf{u}}_{j|i}$ denotes the vote vector from capsule i to capsule j in the next layer

$$\hat{u}_{j|i} = W_j^{c_1} u_i + \hat{b}_{j|i}$$

Zhao et al., 2018

Routing: Dynamic Routing Algorithm

- Unsupervised iterative method for computing routing
- No parameters (But depends on vote vectors)
- Used to connect capsule layers
- Compute next layer of capsules $\{\mathbf{v}_j\}$ from vote vectors

Procedure 1 Routing algorithm.

procedure ROUTING($\hat{\boldsymbol{u}}_{j|i}$, r, l)
    for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
    for r iterations **do**
        for all capsule i in layer l: $\mathbf{c}_i \leftarrow \mathtt{softmax}(\mathbf{b}_i)$ ▷ softmax computes Eq. 3
        for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
        for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \mathtt{squash}(\mathbf{s}_j)$ ▷ squash computes Eq. 1
        for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
    return $\mathbf{v}_j$

Sabour et al., 2017

Types of Capsule Layers

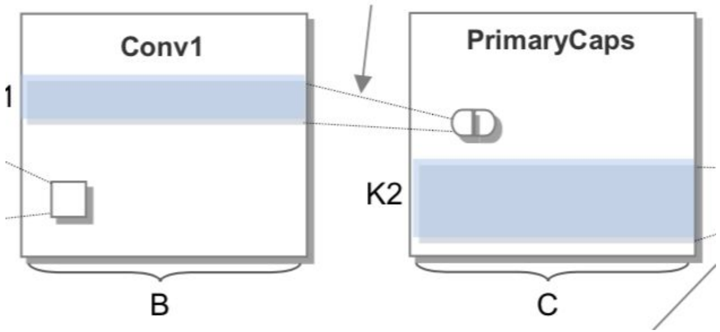
1. **Primary Capsule Layer:** Convolutional output → capsules

2. **Convolutional Capsule Layer:** Local capsules → capsules

3. **Feedforward Capsule Layer:** All capsules → capsules

Primary Capsule Layer

Convolutional output → capsules
Create C capsules from B filters



1. Convolution output with B filters: $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_\mathrm{B}] \in \mathbb{R}^{(L-K_1+1)\times B}$

2. Transform each row of features: $p_i = g(W^b\mathbf{M}_i + \mathbf{b}_1)$ $\quad W^b \in \mathbb{R}^{B\times d}$

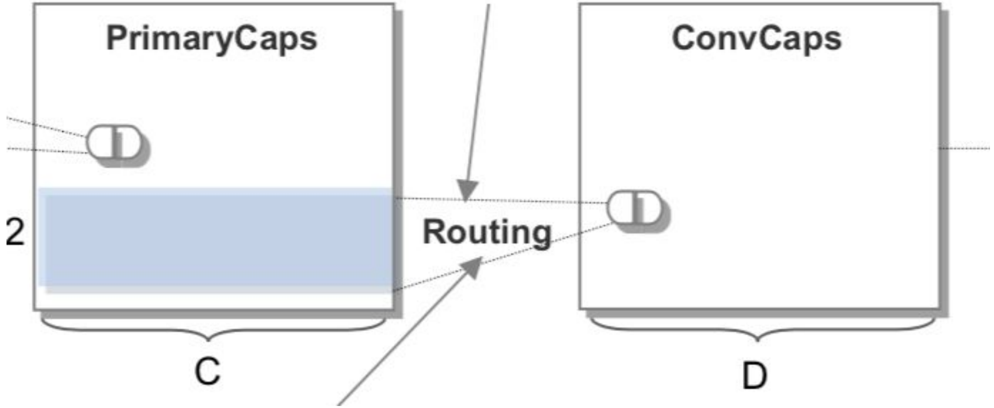
3. Collect C d-dimensional capsules: $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_\mathrm{C}] \in \mathbb{R}^{(L-K_1+1)\times C\times d}$

Zhao et al., 2018

Convolutional Capsule Layer

Local capsules in layer #1 → capsules in layer #2

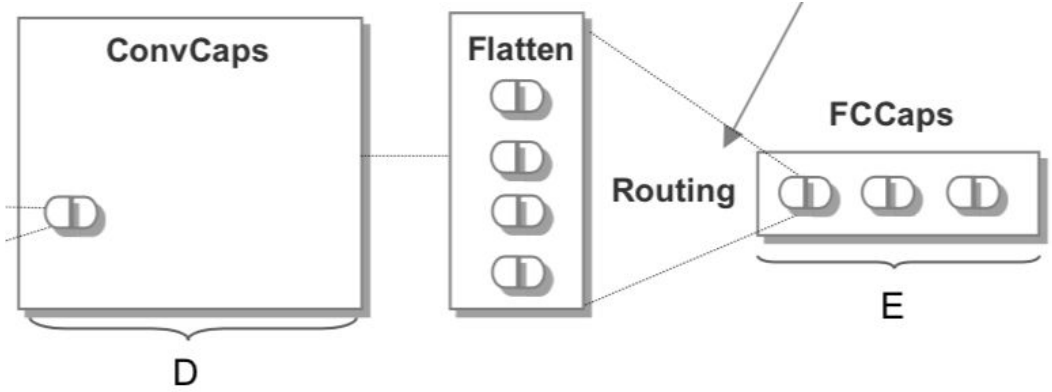
- Route a sliding window of capsules in previous layer into capsules in next layer



Feedforward Capsules Layer

All capsules in layer #1 → capsules in layer #2

1. Flatten all capsules in layer #1 into a vector
2. Route from this vector of capsules into new capsules



Margin Loss

- Identify each output capsule with a class
- Classification loss for capsules
- Calculate on output of feedforward capsule layer
- Ensures that the capsule vector for the correct class is long ($|\mathbf{v}| \approx 1$)

$$L_k = T_k \ \max(0, m^+ - ||\mathbf{v}_k||)^2 + \lambda \, (1 - T_k) \ \max(0, ||\mathbf{v}_k|| - m^-)^2$$

Sabour et al., 2017

Investigating Capsule Networks with Dynamic Routing for Text Classification

Zhao, Ye, Yang, Lei, Zhang, Zhao 2018

Main Ideas

1. Develops capsule network architecture for text classification tasks
2. Achieves state-of-the-art performance on single-class text classification
3. Capsules allow transferring single-class classification knowledge to multi-class task very well

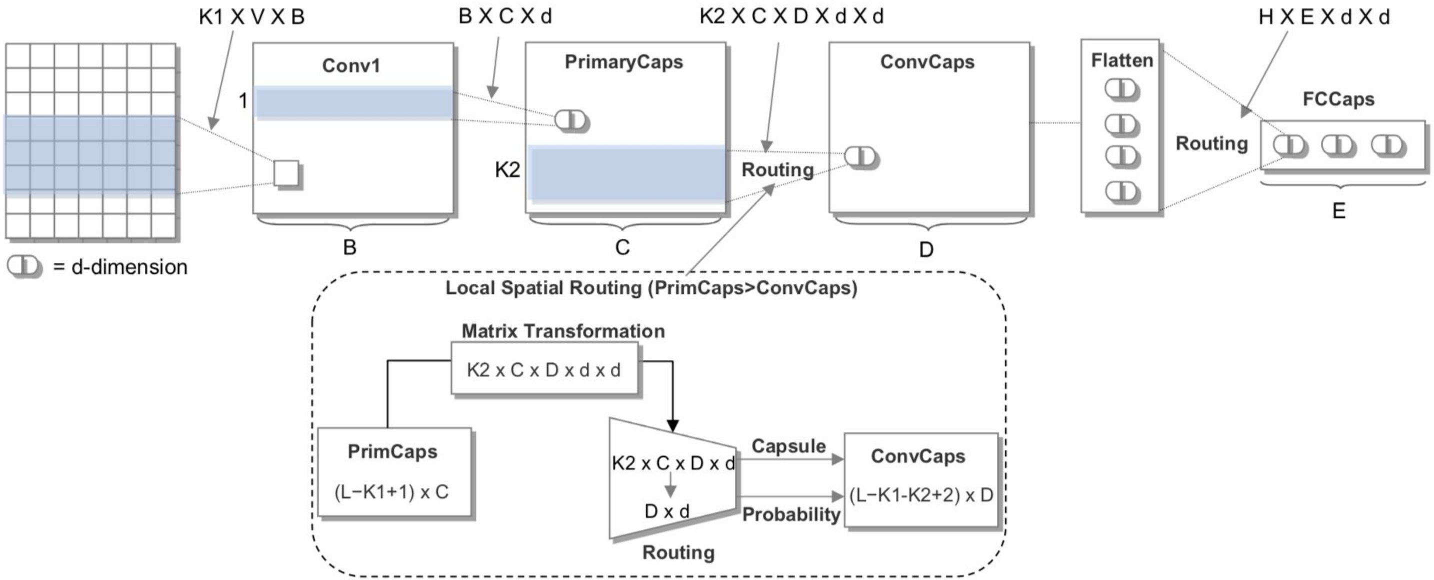
Text Classification

- Read text and classify something about the passage
- Sentiment analysis, toxicity detection, etc.

Multi-Class Text Classification

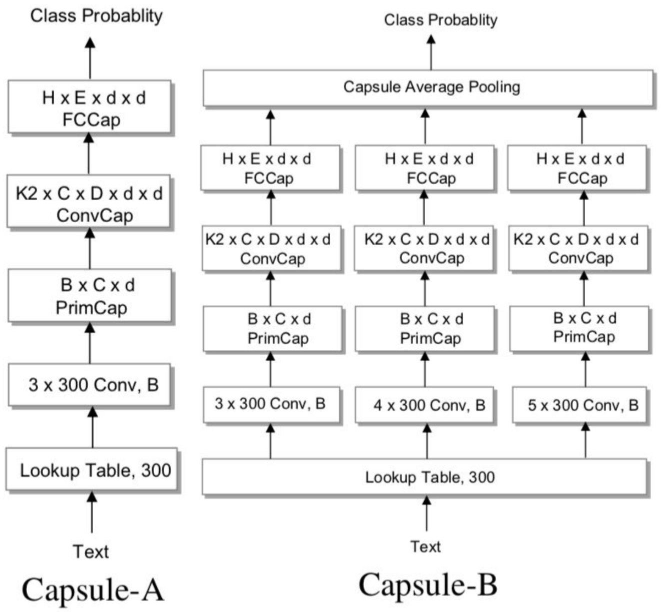
- Document can be labeled as multiple classes
    - Example: In toxicity detection, Toxic and Threatening

Text Classification Architecture



Architectural Variants

- **Capsule-A:** One capsule network
- **Capsule-B:** Three capsule networks that are averaged at the end



Orphan Category

- Add a capsule that corresponds to no class to the final layer
- Network can send words unimportant to classification to this category
  - Function words like *the*, *a*, *in*, etc.
- More relevant in the NLP domain than in images because images don't have a "default background"

Datasets

Single-Label

Dataset	Train	Dev	Test	Classes	Classification Task
MR	8.6k	0.9k	1.1k	2	review classification
SST-2	8.6k	0.9k	1.8k	2	sentiment analysis
Subj	8.1k	0.9k	1.0k	2	opinion classification
TREC	5.4k	0.5k	0.5k	6	question categorization
CR	3.1k	0.3k	0.4k	2	review classification
AG's news	108k	12.0k	7.6k	4	news categorization

Multi-Label

Dataset	Train	Dev	Test	Description
Reuters-Multi-label	5.8k	0.6k	0.3k	only multi-label data in test
Reuters-Full	5.8k	0.6k	3.4k	full data in test

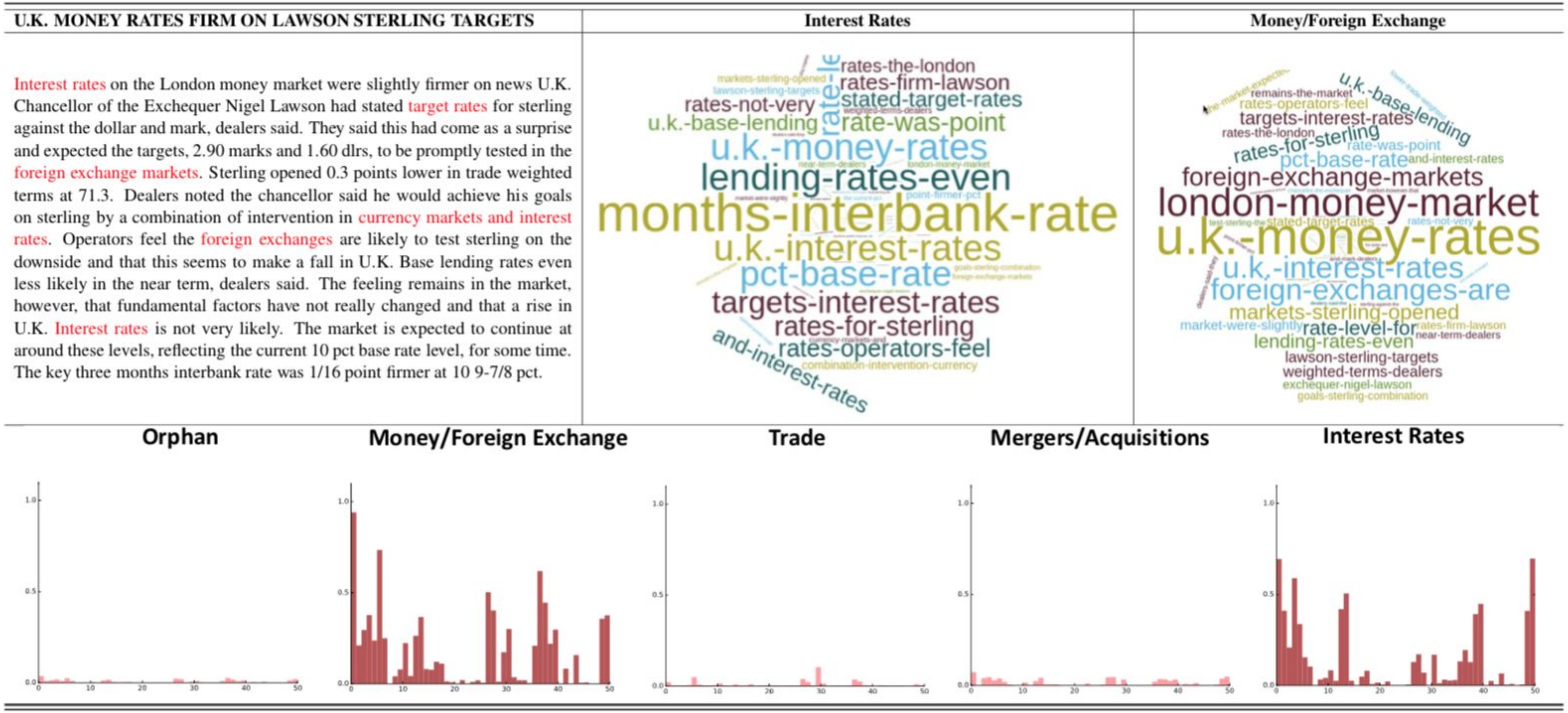
Single-Class Results

	MR	**SST2**	**Subj**	**TREC**	**CR**	**AG's**
LSTM	75.9	80.6	89.3	86.8	78.4	86.1
BiLSTM	79.3	83.2	90.5	89.6	82.1	88.2
Tree-LSTM	80.7	85.7	91.3	91.8	83.2	90.1
LR-LSTM	81.5	**87.5**	89.9	-	82.5	-
CNN-rand	76.1	82.7	89.6	91.2	79.8	92.2
CNN-static	81.0	86.8	93.0	92.8	84.7	91.4
CNN-non-static	81.5	87.2	93.4	**93.6**	84.3	92.3
CL-CNN	-	-	88.4	85.7	-	92.3
VD-CNN	-	-	88.2	85.4	-	91.3
Capsule-A	81.3	86.4	93.3	91.8	83.8	92.1
Capsule-B	**82.3**	86.8	**93.8**	92.8	**85.1**	**92.6**

Multi-Class Transfer Learning Results

	Reuters-Multi-label				Reuters-Full			
	ER	Precision	Recall	F1	ER	Precision	Recall	F1
LSTM	23.3	86.7	54.7	63.5	62.5	78.6	72.6	74.0
BiLSTM	26.4	82.3	55.9	64.6	65.8	83.7	75.4	77.8
CNN-rand	22.5	88.6	56.4	67.1	63.4	78.7	71.5	73.6
CNN-static	27.1	91.1	59.1	69.7	63.3	78.5	71.2	73.3
CNN-non-static	27.4	92.0	59.7	70.4	64.1	80.6	72.7	75.0
Capsule-A	57.2	88.2	80.1	82.0	66.0	83.9	**80.5**	80.2
Capsule-B	**60.3**	**95.4**	**82.0**	**85.8**	**67.7**	**86.4**	80.1	**81.4**

Connection Strength Visualization



Discussion

- Capsule network performs strongly on single-class text-classification
- Capsule model transfers effectively from single-class to multi-class domain
  - Richer representation
  - No softmax in last layer
- Useful because multi-class data sets are hard to construct (exponentially larger than single-class data sets)

Identifying Aggression and Toxicity in Comments Using Capsule Networks

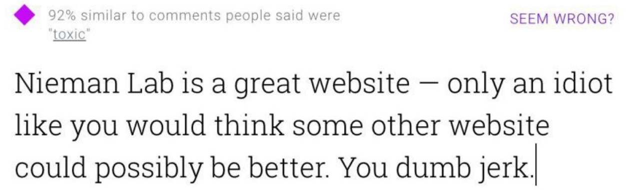
Srivastava, Khurana, Tewari 2018

Main Ideas

1. Develop end-to-end capsule model that outperforms state-of-the-art models for toxicity detection
2. Eliminate need for pipelining and preprocessing
3. Performs especially well on code-mixed comments (comments switching between English and Hindi)

Toxicity Detection

- Human moderation of online content is expensive – useful to do algorithmically
- Classify comments as **toxic**, **severe toxic**, **identity hate**, etc.



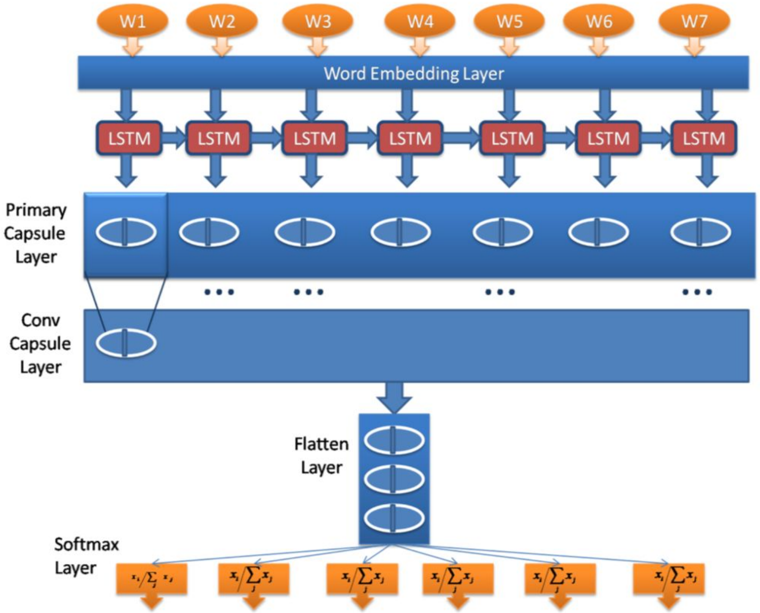
Challenges in Toxicity Detection

- Out-of-vocabulary words
- Code-mixing of languages
- Class imbalance

Why Capsule Networks?

- Seem to be good at text classification (Zhao et al., 2018)
- Should be better at code-mixing than sequential models (build up local representations)

Architecture



- Very similar to architecture to Zhao et al.
- Feature extraction convolutional layer replaced by LSTM
- Standard softmax layer instead of margin loss

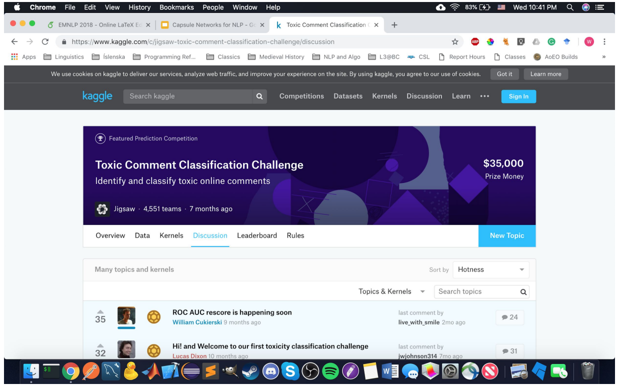
Focal Loss

- Loss function on standard softmax output
- Used to solve the class imbalance problem
- Weights rare classes higher than cross-entropy

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t), \text{ where } p_t = \{ \begin{array}{cc} p & if\ y=1 \\ 1-p & else \end{array}$$

Datasets

- **Kaggle Toxic Comment Classification**
  - English
  - Classes: Toxic, Severe Toxic, Obscene, Threat, Insult, Identity Hate
- **First Shared Task on Aggression Identification (TRAC)**
  - Mixed English and Hindi
  - Classes: Overtly Aggressive, Covertly Aggressive, Non-Aggressive



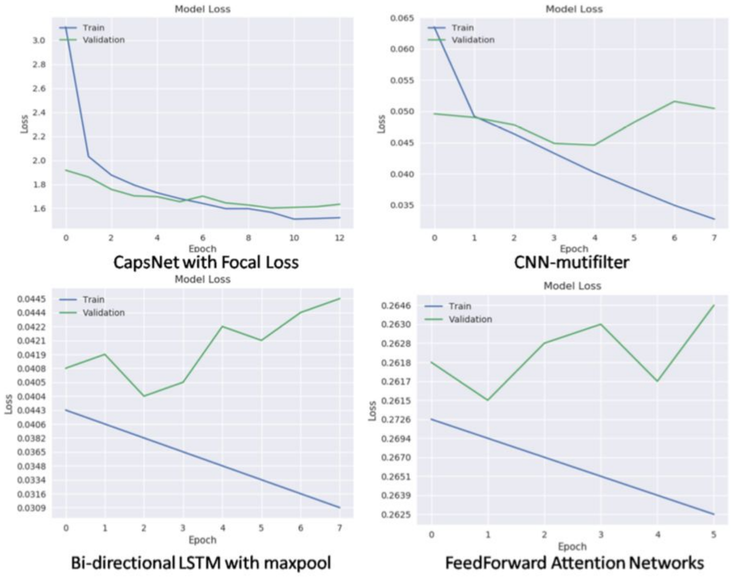
https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion

Results

Model_Name	**Kaggle-toxic comment classification (ROC-AUC)**	**TRAC - 1 (English-FB) (Weighted F1)**	**TRAC - 1 (English-TW) (Weighted F1)**
CNN-multifilter	95.16	55.43	53.41
CNN-LSTM	96.85	62.20	47.68
Bi-directional LSTM with maxpool	97.35	59.79	51.146
FeedForward Attention Networks	97.42	57.43	55.49
Hierarchical ConvNets	97.95	51.38	50.43
Bi-LSTM, Logistic Regression	98.17	57.17	52.1
Bi-LSTM, xgboosted	98.19	57.33	52.31
Bi-LSTM with skip connections	98.20	61.78	51.98
Pre-trained LSTMs	98.25	60.18	58.7
CapsuleNet without Focal Loss	98.21	62.032	58.600
CapsuleNet with Focal Loss	**98.46**	**63.43**	**59.41**

Training/Validation Loss

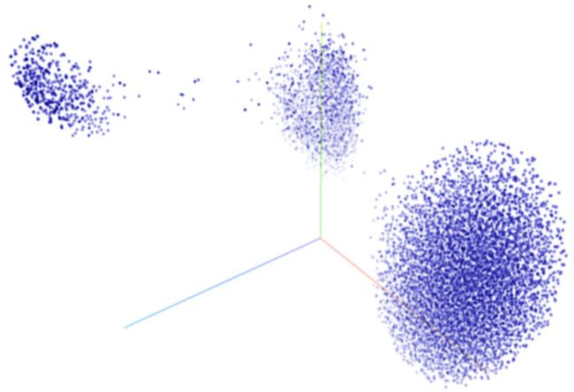
- Training and validation loss stayed much closer for the capsule model
- ⇒ Avoids overfitting



(a) Training and Validation Loss for Kaggle Toxic Comment Classification Dataset

Word Embeddings on Kaggle Corpus

- Three clear clusters:
  - Neutral words
  - Abusive words
  - Toxic words + place names



(b) Clusters for word obtained after training

OOV Embeddings

NN to "politics"	NN to "bharat"
politic	bharatiya
politican	bhar
politico	mahabharata
politicize	bharti
politician	bhaskar

NN to "kut*e"(Hindi)
chu**ya
sa*le
tere
g**d
ma***rc**d

Table 2: Example of handling misspelt words and transliteration. NN : Nearest Neighbour

- Out of vocabulary words randomly initialized
- Converge to accurate vectors

Discussion

- The novel capsule network architecture performed the best on all three datasets
- No data preprocessing done
- Avoids overfitting
- Local representations lead to big gains in mixed-language case

Zero-shot User Intent Detection via Capsule Neural Networks

Xia, Zhang, Yan, Chang, Yu 2018

Main Ideas

1. Capsule networks extract and organize information during supervised intent detection
2. These learned representations can be effectively transferred to the task of zero-shot intent detection

User Intent Detection

- Text classification task for question answering and dialog systems
- Classify which action a user query represents out of a known set of actions
  - **GetWeather**, **PlayMusic**

Zero-Shot User Intent Detection

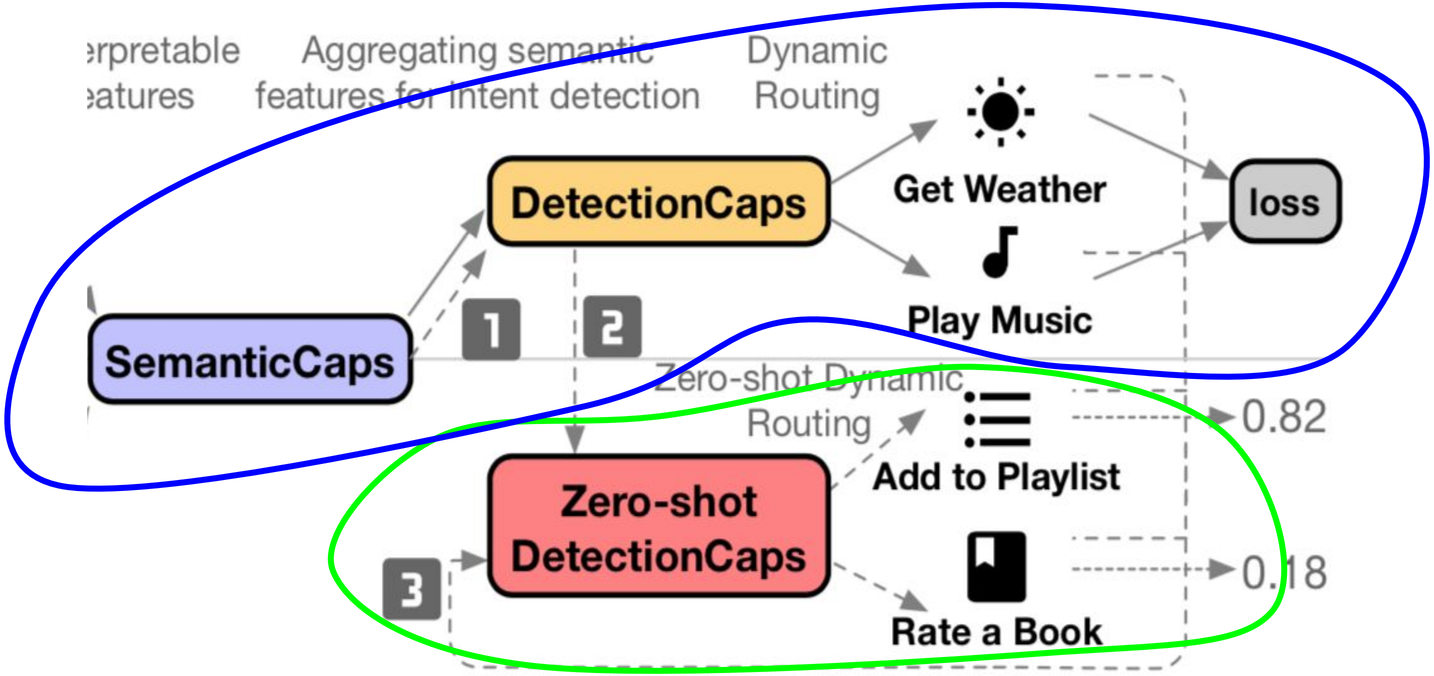
- Training set with known set of intents
  - **GetWeather**, **PlayMusic**
- Test set has unseen “emerging” intents
  - **AddToPlaylist**, **RateABook**
- Transfer information about known intents to new domain of emerging intents

What Signal is There?

- Embedding of the string name of the unknown and known intents
- Output capsules for known intents
- Can combine these two things to do zero-shot learning

Architecture

Network trained on known intents



Extension for zero-shot inference

SemanticCaps Layer

- Extract features using self-attention LSTM

Combine to get **H**

$$\left\{ \begin{aligned} \overrightarrow{\mathbf{h}}_t &= \mathrm{LSTM}_{fw}(\mathbf{w}_t, \overleftarrow{\mathbf{h}}_{t-1}), \\ \overleftarrow{\mathbf{h}}_t &= \mathrm{LSTM}_{bw}(\mathbf{w}_t, \overleftarrow{\mathbf{h}}_{t+1}). \end{aligned} \right\}$$

Self-attention weights

$$\left\{ \mathbf{A} = \mathrm{softmax}\left(\mathbf{W}_{s2} \tanh\left(\mathbf{W}_{s1}\mathbf{H}^T\right)\right) \right\}$$

M is the extracted features

$$\left\{ \begin{gathered} \mathbf{M} = \mathbf{AH} \\ (\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_R) \in \mathbb{R}^{R \times 2D_H} \end{gathered} \right\}$$

DetectionCaps Layer

- Standard convolutional capsule layer → feedforward capsule layer

Loss During Training

- Normal max-margin loss + regularization
- Regularization incentivizes semantic capsules to capture different features
- Regularization controlled by hyperparameter α

Max-margin loss

$$\mathcal{L} = \sum_{k=1}^{K} \{ [\![y = y_k]\!] \cdot \max(0, m^+ - \|\mathbf{v}_k\|)^2 + \lambda\, [\![y \neq y_k]\!] \cdot \max(0, \|\mathbf{v}_k\| - m^-)^2 \}$$

Regularization term

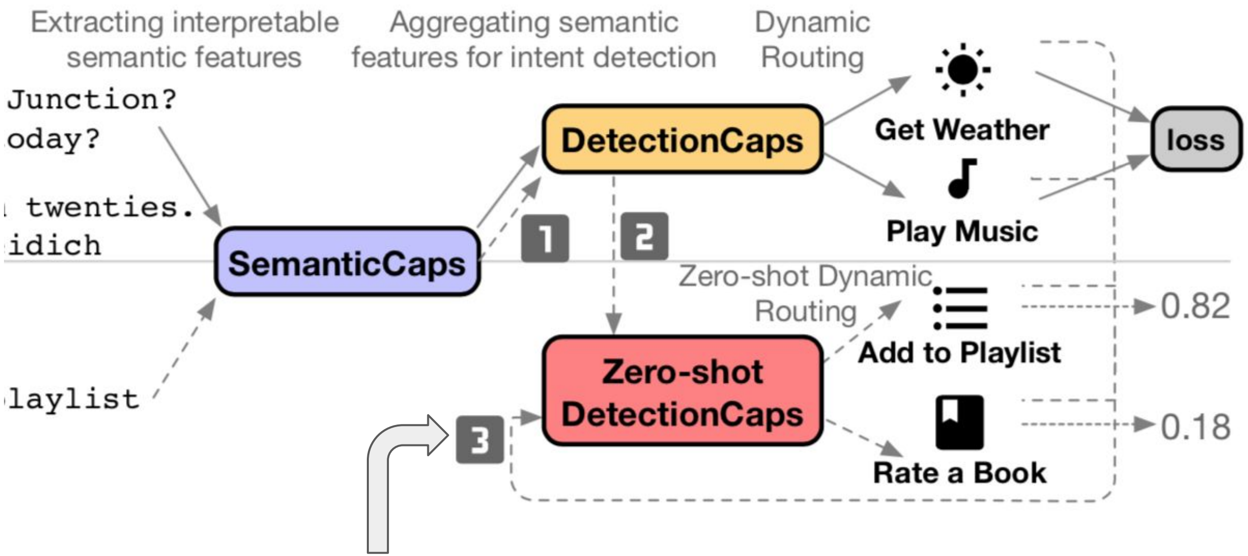
$$+ \alpha ||\mathbf{A}\mathbf{A}^T - I||_F^2,$$

Intent Detection Results

Model	SNIPS-NLU (on 5 existing intents)				CVA (on 80 existing intents)			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
TFIDF-LR	0.9546	0.9551	0.9546	0.9545	0.7979	0.8104	0.7979	0.7933
TFIDF-SVM	0.9584	0.9586	0.9584	0.9581	0.7989	0.8111	0.7989	0.7942
CNN	0.9595	0.9596	0.9595	0.9595	0.8223	0.8288	0.8223	0.8210
RNN	0.9516	0.9522	0.9516	0.9518	0.8286	0.8330	0.8286	0.8275
GRU	0.9535	0.9535	0.9535	0.9534	0.8239	0.8281	0.8239	0.8216
LSTM	0.9569	0.9573	0.9569	0.9569	0.8319	0.8387	0.8319	0.8306
Bi-LSTM	0.9501	0.9502	0.9501	0.9502	0.8428	0.8479	0.8428	0.8419
Self-attention Bi-LSTM	0.9524	0.9522	0.9524	0.9522	0.8521	0.8590	0.8521	0.8513
INTENTCAPSNET	**0.9621**	**0.9620**	**0.9621**	**0.9620**	**0.9088**	**0.9160**	**0.9088**	**0.9023**

Architecture Revisited

- Goal: Use predicted capsules for known intents for zero-shot inference



Generalizing to Emerging Intents

- Build similarity matrix between existing intents and emerging intents based on embeddings for intent names:

$$q_{lk} = \frac{exp\left\{-d\left(\mathbf{e}_{z_l}, \mathbf{e}_{y_k}\right)\right\}}{\sum_{k=1}^{K} exp\left\{-d\left(\mathbf{e}_{z_l}, \mathbf{e}_{y_k}\right)\right\}},$$

where

$$d\left(\mathbf{e}_{z_l}, \mathbf{e}_{y_k}\right) = \left(\mathbf{e}_{z_l} - \mathbf{e}_{y_k}\right)^T \Sigma^{-1} \left(\mathbf{e}_{z_l} - \mathbf{e}_{y_k}\right)$$

Classifying Emerging Intents

1. Goal is to get prediction vector for emerging intent *l*
2. Have vote vectors $\mathbf{g}_{k,r}$ from known intent classification
3. Represent vote vector for emerging intent as weighted sum of known intents:

$$\mathbf{u}_{l|r} = \sum_{k=1}^{K} q_{lk} \mathbf{g}_{k,r}$$

4. Use dynamic routing to get an activation capsule $\mathbf{n}_l$ for each emerging intent
5. Pick the $\mathbf{n}_l$ with largest magnitude

Zero-Shot Intent Detection Results

Model	SNIPS-NLU (on 2 emerging intents)				CVA (on 20 emerging intents)			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
DeViSE (Frome et al., 2013)	0.7447	0.7448	0.7447	0.7446	0.7809	0.8060	0.7809	0.7617
CMT (Socher et al., 2013)	0.7396	**0.8266**	0.7396	0.7206	0.7721	0.7728	0.7721	0.7445
CDSSM (Chen et al., 2016a)	0.7588	0.7625	0.7588	0.7580	0.2140	0.4072	0.2140	0.1667
Zero-shot DNN (Kumar et al., 2017)	0.7165	0.7330	0.7165	0.7116	0.7903	0.8240	0.7903	0.7774
INTENTCAPSNET-ZSL w/o Self-attention	0.7587	0.7764	0.7588	0.7547	0.8103	0.8512	0.8103	0.8115
INTENTCAPSNET-ZSL w/o Bi-LSTM	0.7619	0.7631	0.7619	0.7616	0.8366	**0.8770**	0.8366	0.8403
INTENTCAPSNET-ZSL w/o Regularizer	0.7675	0.7676	0.7675	0.7675	0.8544	0.8730	0.8544	0.8553
INTENTCAPSNET-ZSL	**0.7752**	0.7762	**0.7752**	**0.7750**	**0.8628**	0.8751	**0.8629**	**0.8635**

Discussion

- Representational power of capsule network can be leveraged for zero-shot learning
- Interesting regularizations and architectural extensions for capsule networks

Conclusion

- Capsule representations encode “instantiation parameters” of features
- Papers follow a standard CapsNet architecture for text classification:
    a. Features Extraction (ConvNet or LSTM)
    b. Primary Capsule Layer
    c. Convolutional Capsule Layer
    d. Classification (Margin or softmax)
- Capsule representations can be leveraged for transfer/zero-shot learning

Discussion Questions

1. What is powerful about capsule representations?
2. Are capsule networks good for NLP, or are they just good for vision?
3. Why has NLP capsule research focused on text classification tasks?
4. What are some other NLP tasks that capsule networks could be applied to?
5. What other advanced architectures could be useful in NLP?

Other Papers

- Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869.

Other Materials

- https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc
- https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b