Transformer-Capsule Model for Intent Detection (Student Abstract)

Aleksander Obuchowski,[1,2] **Michał Lew**[2]
[1]Faculty of Applied Physics and Mathematics, Gdańsk University of Technology
[2]SentiOne Research
s174086@student.pg.edu.pl, michal.lew@sentione.com

Abstract

Intent recognition is one of the most crucial tasks in NLU systems, which are nowadays especially important for designing intelligent conversation. We propose a novel approach to intent recognition which involves combining transformer architecture with capsule networks. Our results show that such architecture performs better than original capsule-NLU network implementations and achieves state-of-the-art results on datasets such as ATIS, AskUbuntu ,and WebApp.

Introduction

Given the increasing presence of chatbots and digital assistants in our daily lives, the demand for conversation systems is especially high. Natural Language Understanding (NLU) is a crucial component in designing those systems. Intent detection is a part of NLU that focuses on capturing intention of user queries. Given the input sentence, its goal is to assign it to a specific label, that can later be used by the conversation system to return an adequate answer. Machine learning models designed for this task usually consist of an encoder and a classifier. The encoders role is to create semantically accurate embedding of the input text that can further be used by the classifier to assign a label to the query.

There are many approaches to producing embeddings for the whole sentence from word-level embeddings. One of the basic approaches is to use LSTM networks (Gers, Schmidhuber, and Cummins 1999) which, through their recurrent structure, accurately capture the contextual relations of words in a sentence. Next, LSTM with attention (Bahdanau, Cho, and Bengio 2014) was used, where attention mechanism enabled the encoder to focus on the important words in the sentence. Recently one of the most popular architectures used in sentence encoding has been transformer architecture (Vaswani et al. 2017) that uses just the attention mechanism, without recurrent connections.

Capsule neural networks, primary used in image recognition (Sabour, Frosst, and Hinton 2017) are new types of neural networks that group neurons together into vectors that encode specific parameters of entities and use dynamic-routing between layers to pass on the parameters that are important onto the next layer. Capsule networks have also been used in NLP (Zhang et al. 2018) achieving high accuracy on intent recognition task.

In our solution we decided to combine both transformer and capsules architecture. This was motivated by the fact that transformers, through their attention mechanism, do en excellent job in encoding short pieces of text focusing on important words, while capsule networks, thought their dynamic routing, enable propagation of important features through other layers of the network.

We have tested our solution on AskUbuntu and WebApp where we achieved results of 89% and 92%, outperforming current state-of-the-art solution (Shridhar et al. 2018), as well as on ATIS dataset where we achieved 98.89% with current state-of-the-art (Chen and Yu 2019) at 98.61%.

Approach

In our approach we used the encoder part of transformer architecture combined with our implementation of capsule networks and their dynamic routing to construct an accurate embedding for the queried sentence. The whole architecture is shown in Figure1.

As a input we used GloVe (Pennington, Socher, and Manning 2014) embeddings pre-trained on Common Craw corpus. Those inputs were fed to a transformer module (Figure 1). This module consisted of a single layer transformer encoder with 12-head attention and feed-forward layer with hidden dimension of 300.

Vector produced by the transformer was then fed into capsule module (Figure 1) consisting of 100 capsules with 15 dimensions each. Dynamic routing was performed 4 times. We found this number to be optimal for the ensuring selection of important features while still considering those that are less aligned with the output. Then, inline with the original capsules architecture, we used squash function of the capsules output. Next, we used flatten layer to reduce the number of dimensions to be analyzed by further layers.

Finally, a sentence encoded by this model was fed into a dense layer with softmax activation function that mapped the embedded vector to the desired class in one-hot format. We used cross-entropy as loss function and adam optimizer (Kingma and Ba 2014).

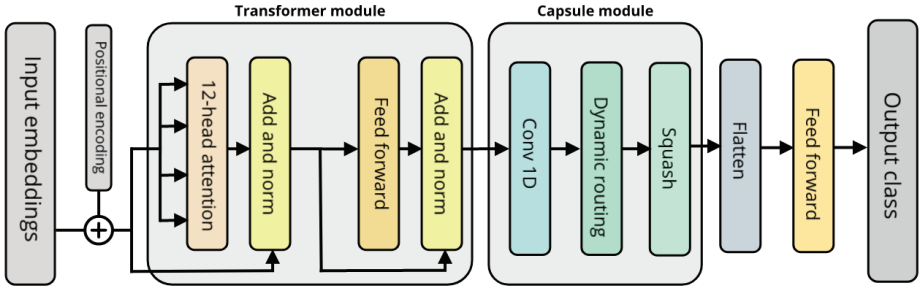

Figure 1: Our architecture

Experiments and Results

For our experiments, we used the following datasets: AskUbuntu, WebApp (Braun et al. 2017), and ATIS.

On AskUbuntu and WebApp we compared our solution with popular architectures [1] as well as current state-of-the-art solution on those datasets - Subword Semantic Hashing (SSH) (Shridhar et al. 2018).

	AskUbuntu	WebApp
Botfuel	0.90	0.80
Luis	0.90	0.81
API (DialogFlow)	0.85	0.80
Watson	0.92	0.83
RASA	0.86	0.74
Snips	0.83	0.78
Recast	0.86	0.75
SSH	0.94	0.85
Transformer-Capsule	**0.98**	**0.92**

Table 1: Comparison of different intent recognition services (micro f1 score)

We also compared our solution with the original capsule implementation (Zhang et al. 2018) that was tested on the ATIS dataset as well as the current SOTA on this dataset (Chen and Yu 2019).

	ATIS
CAPSULE-NLU	0.950
WAIS	0.9861
Tranformer-Capsule	**0.9889**

Table 2: Comparison of our model to capsule-nlu and SOTA on ATIS dataset (accuracy score)

The results show that our solution is able to achieve state-of-the-art results on datasets with small number of examples, such as AskUbuntu and WebApp, as well as larger datasets like ATIS.

Conclusion and Future Work

Using transformer encoder with the capsule architecture can lead to better results in intent recognition task. For our future work we are planning to test this architecture on joint intent recognition and slot-filing, as well as explore how pre-training of the encoder on larger corpora can improve the results.

[1] https://github.com/Botfuel/benchmark-nlp-2018

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Braun, D.; Hernandez-Mendez, A.; Matthes, F.; and Langen, M. 2017. Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 174–185. Saarbrücken, Germany: Association for Computational Linguistics.

Chen, S., and Yu, S. 2019. Wais: Word attention for joint intent detection and slot filling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9927–9928.

Gers, F. A.; Schmidhuber, J.; and Cummins, F. 1999. Learning to forget: Continual prediction with lstm.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic routing between capsules. In *Advances in neural information processing systems*, 3856–3866.

Shridhar, K.; Sahu, A.; Dash, A.; Alonso, P.; Pihlgren, G.; Pondeknath, V.; Simistira, F.; and Liwicki, M. 2018. Subword semantic hashing for intent classification on small datasets. *arXiv preprint arXiv:1810.07150*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Zhang, C.; Li, Y.; Du, N.; Fan, W.; and Yu, P. S. 2018. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.