AUXILIARY CAPSULES FOR NATURAL LANGUAGE UNDERSTANDING

Ieva Staliūnaitė, Ignacio Iacobacci

Huawei Noah's Ark Lab, London, United Kingdom

ABSTRACT

Lately, joint training of Intent detection and Slot filling has become the best-performing approach in the field of Natural Language Understanding (NLU). In this work we extend the newly introduced application of Capsule Networks for NLU to a multi-task learning environment, using relevant auxiliary tasks. Specifically, our models perform joint Intent classification and Slot filling with the aid of Named Entity Recognition (NER) and Part of Speech (POS) tagging tasks. This allows us to exploit the hierarchical relationships between the Intents of the utterances and the different features of input text, not only Slots but also Named Entity mentions, Parts of Speech, quantity indications, etc. The models developed in this work are evaluated on standard benchmarks, achieving state-of-the-art results on the SNIPS dataset while outperforming the best commercial systems on several low-resource datasets.

Index Terms— natural language understanding, multi-task learning, capsule networks, intent recognition, slot filling

1. INTRODUCTION

The initial task in pipeline conversational dialogue systems is Natural Language Understanding (NLU), which is essentially the task of interpreting user inputs. NLU is a key segment of dialogue systems as the errors produced in this early step propagate and increase in the follow-up processing elements, hurting the capabilities of the whole system. As pointed out by Gao et al. [1], the responsibility of NLU in dialogue systems is threefold: it performs **Domain detection**, **Intent classification** and **Slot filling**. In this work we tackle the two latter tasks. To illustrate, an NLU system interprets the utterance in the example (1) below as having the Intent 'book restaurant' and the Slots 'party size description' and 'restaurant type', whose values are marked in bold.

(1) **my great grandfather and I** would like to get together at a **taverna**

As example (1) shows, the tasks of recognizing the Intent and the Slot are not independent. That is, the fact that the utterance provides information about a size of a party and a type of restaurant are useful clues for the fact that the sentence was used to book a restaurant rather than rate a film.

NLU tasks are widely studied in the NLP community and the state of the art in NLU is currently achieved by neural-network-based models [2, 3]. Transfer learning [4, 5, 6] and multi-task [7, 8, 9, 10] methods are commonly used for Slot filling and Intent classification in order to use the knowledge that can be learned from other tasks to improve the results on the target tasks. In transfer learning approaches models are pre-trained, generally in an unsupervised manner, and later finetuned on a specific task. In contrast, in multi-task approaches, a single model is trained to generate the output for multiple related (target and auxiliary) objectives with the aim to generalize from the linguistic information provided by the annotations of the different tasks. Various types of auxiliary annotations are used for both transfer and multi-task learning, from POS tags to NER tags and semantic tags. They can enrich the NLU models with relevant information, aiding the classification by making use of correlations between proper nouns and destinations, superlatives and relative costs, etc.

The current state-of-the-art model on the SNIPS dataset[1] by Zhang et al. [2] uses Capsule Networks [11, 12] that are introduced in section 3. Previously, Renkens et al. [13] applied Capsule Networks to Spoken Language Understanding (SLU), and Zhang et al. [2] had adapted it to NLU. Our contribution is twofold: (i) we introduce a Capsule Network-based approach that leverages auxiliary tasks to perform joint Intent classification and Slot filling, and (ii) we present experimental results which show that such models encode hierarchical relationships that aid classification not only between the target tasks but also NER and POS tagging.

2. RELATED WORK

Research in the area of NLU has recently focused on methods that make use of the relationships between Slots and Intents [2, 14, 15] as it has been shown that joint training is beneficial to both tasks [16]. Zhao et al. [14] developed a hierarchical encoder-decoder model for the two tasks. Their model encodes the input utterance into a latent representation, which is later decoded in a hierarchical way into *(Intent, Slot_type, Slot_value)* triples. Goo et al. [15] present a different approach wherein a Slot gate is used as a weighted feature that indicates whether the Intent and Slot predictors pay atten-

[1] https://github.com/snipsco/nlu-benchmark

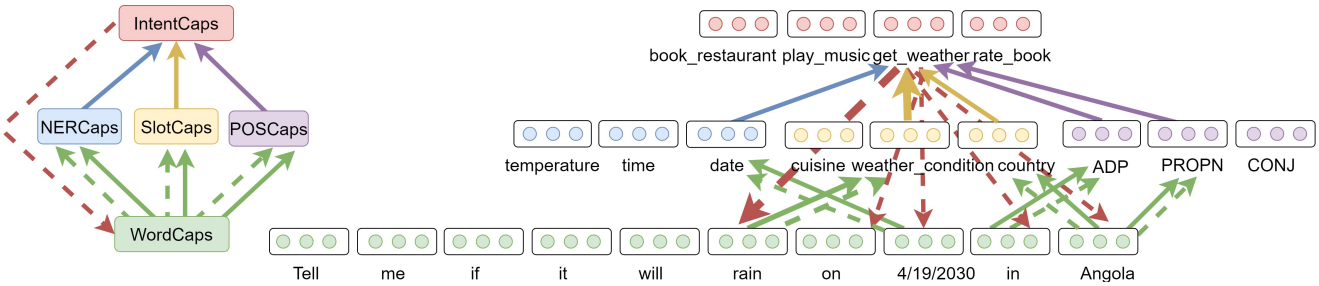

Fig. 1. Illustration of the model adapted from Capsule-NLU [2]. The solid arrows mark the dynamic routing paths and the dashed arrows stand for the re-routing paths. The proposed model adds NERCaps and POSCaps to the original SlotCaps by concatenation in order to make use of additional linguistic information in the input.

tion to the same elements in the input sequence. In turn, the NLU model of Gupta et al. [3] only shares weights between the two tasks in a word contextualization step and reaches the highest scores for the two tasks measured independently.

Based on the Semantic Error[2] metric the current state-of-the-art results on ATIS and SNIPS belong to Haihong et al. [17] and Zhang et al. [2], respectively. Haihong et al. [17] present SF-ID, a model which consists of a Slot filling subnet and an Intent detection subnet, the two sharing the information between themselves about the predicted Slots and Intents. In contrast, Zhang et al. [2] use Capsule Networks to perform Intent classification and Slot filling. Prior to the latter work, Xia et al. [18] had reached the state of the art of Intent classification using a Capsule Network with only words as lower level features and Intents as higher level features. Hence, Capsules appear to have high capacity to capture the aforementioned relationship between lower level features and higher level features in NLU.

3. MODEL

The novel type of Neural Network layers of the Capsule Network offers the possibility to encode hierarchical relationships between lower level and higher level features, in addition to making the model invariant to features that are irrelevant to the task at hand (instantiation parameters) [11]. That is achieved by the shape of the Capsule – in contrast to the neuron used in traditional Neural Networks, a Capsule substitutes the scalar neuron with a vector shape. One Capsule per class is used for classification. The length of the Capsule represents the probability of a respective class, whereas the dimensions of the vector encode the instantiation features of the example of the class. A dynamic routing mechanism is used for measuring agreement between the predictions of the lower level features and the higher level features, consequently biasing the model towards predicting the features which have a better coupling between the features at different levels [12].

[2]Semantic Error is the strictest metric that evaluates both tasks together – it represents the percentage of the test set sentences for which all the Slots as well as the Intent have been correctly predicted.

Our models are built upon the Capsule-NLU [2] model, which adapts the Capsule Network to NLU and enhances it with a re-routing mechanism which essentially allows the information to be passed not only from the predictions of the Slots (lower level features) to Intents (higher level features) but also vice versa. We combine the Capsule Network architecture with the auxiliary tasks of POS tagging and NER in a multi-task learning procedure. The model inputs a set of word embeddings which are connected to a BiLSTM layer. The intermediate states of the BiLSTM constitute the WordCaps. The WordCaps are fed to three Capsule layers: SlotCaps, NERCaps and POSCaps, which provide predictions for Slot values, Named Entities and POS tags. The three Capsule layers are then concatenated and passed to the Intent Capsule.

Fig. 1 illustrates the model with an example from the SNIPS dataset. The dynamic routing mechanism enables the model to learn the relations between the words and all three types of tags, as well as between the tags and the Intent of the sentence.In addition, we add a natural extension to the re-routing mechanism introduced in [2] so that the model learns to discern which words are relevant to the three tagger Capsules (corresponding to Slots, Entities and POS tags) in the middle layer given the predicted Intent of the sentence. The dynamic routes (solid arrows) in the figure show that the model learns to associate the word ‘rain’ with the Slot ‘weather condition’, the numerical value ‘4/19/2030’ with the ‘date’ Entity, the preposition ‘in’ with the POS tag ‘ADP’, etc. In turn, the values in the middle layer of the model which are associated to the input words have higher probability than unrelated Slots such as ‘cuisine’, and are themselves associated with the Intent ‘get weather’. The re-routing mechanism (dashed arrows) enhances the probability of predicting the Slot ‘weather condition’ via the predicted Intent ‘get weather’ and its relation to the input words.

We build additional models for ablation purposes. Firstly, besides the full Capsule ISNP model with four tasks (**I**ntent recognition, **S**lot filling, **N**amed Entity Recognition and **P**OS tagging) we also train and test models with only one of the auxiliary tasks as well as no auxiliary tasks (Capsule ISN, Capsule ISP and Capsule IS). Secondly, in order to compare

the performance of the Capsule Network-based models to other types of models, we build baseline Joint models (Joint ISNP, Joint ISN, Joint ISP, Joint IS). These models use a BiLSTM layer with shared weights between all 4 tasks and additional Conditional Random Fields (CRF) layers for each sequence tagging task (Slot filling, Duckling NER and POS tagging), while a softmax layer is used for Intent prediction. Thirdly, for the extremely low-resource datasets, we train and test the respective models using ELMo [19] as our set of pretrained contextualized word embeddings.

4. EXPERIMENTAL SETUP

4.1. Data

For performance comparison purposes, we evaluate our models on five different datasets. Firstly, we benchmark on the two most commonly used datasets, namely ATIS [20] which consists of flight booking questions and requests (5,871 sentences, 21 Intents, 120 Slots), and SNIPS[3] which covers a wider range of tasks such as playlist updates, weather forecasts, etc. (14,484 sentences, 7 Intents, 72 Slots).

In addition, due to the fact that Capsule Networks have been shown to be able to generalize well from smaller amounts of data than traditional Neural Networks, three very low-resource datasets were used for evaluating the models as well, namely AskUbuntu[4] (190 sentences, 5 Intents, 3 Slots), WebApps[5] (100 setences, 8 Intents, 3 Slots) and Chatbot (206 sentences, 2 Intents, 5 Slots), introduced in [21]. Furthermore, removing the items that contain the most common Intent[6] from ATIS yields a sub-dataset (ATIS^{sub}) containing 1,537 examples, which we experiment on in order to see the effects of auxiliary Capsules in a low-resource setting on ATIS.

4.2. Implementation Details

Some external libraries were used in our experiments. For splitting the input sentences we use the Spacy[7] tokenizer. Part of speech tagging is also performed using Spacy, while Duckling[8] is used for recognizing specific types of Entities, including time references, distances, temperatures, ordinals, etc.

We perform a hyperparameter search in order to find the best configurations for the models, evaluating them on the validation set. The hyperparameter search space is defined by the following values: layer size [1024, 512], number of routing iterations [2, 3, 4], margin loss [0.1, 0.2], Capsule dimensionality [128, 256]. We evaluate our work with the F1, Accuracy and Semantic Error metrics (c.f. footnote 2) and report the scores on the test set by the models with the configurations that yielded the highest validation scores.

[3]https://github.com/snipsco/nlu-benchmark
[4]https://askubuntu.com
[5]https://webapps.stackexchange.com/
[6]The Intent 'flight' appears in 74% of the examples in ATIS.
[7]https://spacy.io/
[8]https://duckling.wit.ai/

5. RESULTS

The results of the experiments as shown in Table 1 and Table 2 provide evidence that auxiliary Capsule models outperform previous state-of-the-art models on some of the datasets, as well as the fact that the models gain improvements from relevant auxiliary tasks and contextual word embeddings. While the auxiliary Capsule models do not outperform the previous state-of-the-art model (**SF-ID** [17]) on ATIS, they do outperform the state of the art (**Capsule-NLU** [2]) on SNIPS[9]. While **Label-recurrent** [3] reach higher scores on the two tasks evaluated independently on SNIPS than either of the Capsule methods, they do not report the stricter Semantic Error value, which captures the ability of the model to generalize to both tasks. Furthermore, in the low-resource setting (c.f. Table 2), the auxiliary Capsule models with contextual word embeddings outperform some of the widely used industry services for NLU (LUIS[10], Watson[11], API.AI[12], RASA[13]) as reported by Braun et al. [21].

Capsule models outperform the baseline Joint models on most datasets, including SNIPS, WebApps and Chatbot. For the datasets of SNIPS and Chatbot, the fact that Capsule models work so well can be accounted for by the fact that the Slots and Intents are very closely related. That is, the presence of a Slot 'playlist' in SNIPS indicates with near certainty that the Intent is 'add to playlist'. Similarly, in the Chatbot dataset, there are only two Intents ('find connection' and 'departure time') and they can be easily discerned based on whether the question contains a 'criterion' Slot. That is because most of the utterances with the 'departure time' Intent inquire about the time of the *next* ('criterion') train or the *earliest* ('criterion') train, etc. This is different from the ATIS dataset, for example, in which most of the Slots ('city name', 'departure date') appear in utterances with various Intents, such as 'flight fair' or 'airline'. For the WebApps dataset the situation is different, as except for two instances the dataset only contains one Slot ('web service') which is in some way related to each Intent. The fact that Capsule networks perform better on this dataset than the joint models suggests that the Intent Capsule makes use of the instantiation parameters of the single Slot, which are different depending on whether the user inquires about a *password* on a web service or *spam* from a web service.

[9]The Capsule-NLU and Capsule IS results should be comparable, as Capsule IS is based on Capsule-NLU, yet we were not able to reproduce the results of Capsule-NLU [2] on ATIS with their released code and our attempts to reach the authors for consultation have not been successful.
[10]https://www.luis.ai/
[11]https://www.ibm.com/watson/
[12]https://www.api.ai/
[13]https://www.rasa.ai/

Model	ATIS			SNIPS		
	Slot F1	Intent Acc	Semantic Error	Slot F1	Intent Acc	Semantic Error
Capsule-NLU	91.8	97.3	80.9	95.2	95.0	83.4
SF-ID	**95.8**	97.8	**86.9**	92.2	97.4	80.6
Label-recurrent	95.5	**98.1**	-	**93.1**	**98.3**	-
Joint IS	94.6	96.2	83.8	91.6	97.6	80.6
Joint ISN	94.7	96.0	82.5	91.1	98.1	78.9
Joint ISP	94.2	96.4	82.9	92.0	97.1	81.3
Joint ISNP	94.1	95.5	81.4	91.2	97.7	79.4
Capsule IS	94.2	80.6	71.0	92.5	98.0	84.6
Capsule ISN	94.3	85.7	74.9	**93.1**	98.0	**85.6**
Capsule ISP	94.5	90.8	78.7	92.5	97.6	84.0
Capsule ISNP	94.4	89.0	78.1	92.9	98.0	85.0

Table 1. Results of the best-performing models.

Model	WebApps		AskUbuntu		Chatbot	
	Slot F1	Intent F1	Slot F1	Intent F1	Slot F1	Intent F1
LUIS	62.4	81.4	83.7	88.3	92.7	98.1
Watson	50.0	**83.1**	67.1	**91.7**	64.8	97.2
API.AI	0	80.3	67.6	85.3	53.4	92.7
RASA	45.9	78.3	74.5	47.8	92.7	98.1
Joint IS	79.7	66.4	**86.8**	88.6	97.3	97.1
Joint ISN	76.5	67.1	82.2	86.2	96.9	96.2
Joint ISP	75.9	64.6	82.8	86.8	96.1	97.1
Joint ISNP	79.1	66.1	81.9	87.2	97.7	98.1
Capsule IS	83.2	59.7	79.2	80.5	97.2	100
Capsule ISN	**85.3**	66.9	78.4	76.8	97.4	100
Capsule ISP	84.2	52.6	81.9	86.0	97.2	100
Capsule ISNP	82.0	71.8	80.0	79.1	**98.2**	**100**

Table 2. Results on low-resource datasets.

Model	ELMo	$ATIS^{sub}$		
		Slot F1	Intent Acc	Semantic Error
Capsule IS	✗	79.5	48.6	36.4
Capsule ISN	✗	82.1	58.7	49.4
Capsule ISP	✗	80.6	78.9	55.5
Capsule ISNP	✗	78.4	84.2	55.9
Capsule IS	✓	82.1	58.7	49.4
Capsule ISN	✓	82.2	70.0	55.9
Capsule ISP	✓	82.2	82.2	64.8
Capsule ISNP	✓	84.3	84.6	67.6

Table 3. Results on a subset of the ATIS dataset after removing examples that have the 'flight' Intent.

5.1. Ablation Analysis

The use of Duckling NER tags improves the performance of the models, in particular in the instances wherein numerical data (such as weather temperatures, prices, distances, etc.) is relevant to the task at hand. That is, Duckling helps predicting the Slots correctly for both SNIPS and ATIS datasets. For example, in SNIPS, the recognition of the word 'four' by Duckling as a 'cardinal' Entity in the example (2) below helps the Capsule ISN model correctly classify it as a 'rating value', whereas Capsule IS mistakenly classifies it as 'entity name'.

(2) put four rating on the raging quiet

In ATIS, Duckling tags are also beneficial for Intent prediction, as mentions of times, prices and distances are predictive of what information the user seeks, e.g. if the user mentions a temporal Entity, they are not likely asking for the time of the flight.

The POS auxiliary task gives a larger boost to model performance than the Duckling NER tagger, as it contains more varied information about the input text. The POS tagger captures the prepositions that might indicate a following location, proper nouns that are likely Named Entities, as well as numerical values. The contributions of the POS tagger are most obvious in ATIS and AskUbuntu, and interestingly the effect remains present even when contextual embeddings are used, which suggests that not all syntactic information can be deduced from these word embeddings.

Finally, the results on the sub-dataset $ATIS^{sub}$ show the same pattern of improvement as the other datasets. Namely, auxiliary tasks as well as contextualized word embeddings increase model performance here as well (see Table 3). Notably, having the auxiliary task of POS tagging, the model performance increases by 52% on the Semantic Error metric.

6. CONCLUSION

By and large, this paper presents the first work combining Capsule Network layers with auxiliary tasks for NLU. The experiments conducted show that this approach is promising as it yields state-of-the-art results on one of the most widely used datasets for this task (SNIPS) as well as a very low-resource dataset (Chatbot). Two conclusions can be drawn on the basis of the results of this work: (1) Capsule networks can be successfully adapted to a multi-task environment with auxiliary tasks; (2) the possibility of exploiting hierarchical relationships between the Intents and different features is a key factor for improving Intent classification. Novel architectures for Capsule networks could be explored in future work in order to find the best ways to combine the Capsules and the features they encode. Furthermore, the high performance of the Capsule models in NLU suggests that the use of Capsule networks could be extended to other tasks in NLP where the interaction between lower level features and higher level features is relevant to the task at hand.

7. REFERENCES

[1] Jianfeng Gao, Michel Galley, and Lihong Li, "Neural approaches to conversational ai," *Foundations and Trends in Information Retrieval*, vol. 13, no. 2-3, pp. 127–298, 2019.

[2] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu, "Joint Slot Filling and Intent Detection via Capsule Neural Networks," in *Proceedings of the 57th ACL*, Florence, Italy, 2019, pp. 5259–5267.

[3] Arshit Gupta, John Hewitt, and Katrin Kirchhoff, "Simple, fast, accurate intent classification and slot labeling for goal-oriented dialogue systems," in *Proceedings of the SIGdial*, Stockholm, Sweden, 2019, pp. 25–34.

[4] Minwoo Jeong and Gary Geunbae Lee, "Multi-domain spoken language understanding with transfer learning," *Speech Communication*, vol. 51, no. 5, pp. 412–424, 2009.

[5] Sungjin Lee and Rahul Jha, "Zero-shot adaptive transfer for conversational language understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, 2019, vol. 33, pp. 6642–6649.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the NAACL-HLT*, Minneapolis, Minnesota, 2019, vol. 1, pp. 4171–4186.

[7] Samuel Louvan and Bernardo Magnini, "Leveraging non-conversational tasks for low resource slot filling: Does it help?," in *Proceedings of the SIGdial*, Stockholm, Sweden, 2019, pp. 85–91.

[8] Zhilin Yang, Ruslan Salakhutdinov, and William W. Cohen, "Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks," in *Proceedings of 5th ICLR*, Toulon, France, 2017.

[9] Juan Diego Rodriguez, Adam Caldwell, and Alexander Liu, "Transfer learning for entity recognition of novel classes," in *Proceedings of the 27th COLING*, Santa Fe, New Mexico, 2018, pp. 1974–1985.

[10] Samuel Louvan and Bernardo Magnini, "Exploring named entity recognition as an auxiliary task for slot filling in conversational language understanding," in *Proceedings of SCAI workshop at EMNLP*, Brussels, Belgium, 2018, pp. 74–80.

[11] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang, "Transforming auto-encoders," in *International Conference on Artificial Neural Networks*, Berlin, Heidelberg, 2011, Springer, pp. 44–51.

[12] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, Long Beach, California, 2017, pp. 3856–3866.

[13] Vincent Renkens and Hugo Van Hamme, "Capsule networks for low resource spoken language understanding," in *Interspeech*, Hyderabad, India, 2018, pp. 601–605.

[14] Zijian Zhao, Su Zhu, and Kai Yu, "A hierarchical decoding model for spoken language understanding from unaligned data," in *ICASSP 2019*, Brighton, UK, 2019, pp. 7305–7309.

[15] Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proceedings of the 2018 Conference of the NAACL-HLT*, New Orleans, Louisiana, 2018, vol. 2, pp. 753–757.

[16] Dilek Hakkani-Tür, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Interspeech*, San Francisco, California, 2016, pp. 715–719.

[17] E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *Proceedings of the 57th ACL*, 2019, pp. 5467–5471.

[18] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu, "Zero-shot user intent detection via capsule neural networks," in *Proceedings of the 2018 EMNLP*, Brussels, Belgium, 2018, pp. 3090–3099.

[19] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the NAACL-HLT*, New Orleans, Louisiana, 2018, vol. 2.

[20] Charles T Hemphill, John J Godfrey, and George R Doddington, "The atis spoken language systems pilot corpus," in *Proceedings of the Speech and Natural Language Workshop*, 1990.

[21] Daniel Braun, Adrian Hernandez-Mendez, Florian Matthes, and Manfred Langen, "Evaluating natural language understanding services for conversational question answering systems," in *Proceedings of the SIGdial*, Saarbrücken, Germany, 2017, pp. 174–185.