First Step Towards Enhancing Word Embeddings with Pitch Accent Features for DNN-based Slot Filling on Recognized Text

Sabrina Stehwien, Ngoc Thang Vu

Universität Stuttgart
{sabrina.stehwien, thang.vu}@ims.uni-stuttgart.de

Abstract: Slot filling, as a subtask of spoken language understanding, is designed to extract key query terms from text after it has been recognized from speech. Most state-of-the-art models do not, however, take recognition error into account and show a substantial drop in performance when applied to recognized text. One source of information that marks important parts of utterances and is available from speech data is prosody. Since pitch accents have been shown to correlate with semantic slots in the ATIS benchmark corpus, we combine these as features with word embeddings for slot filling on ATIS and compare their impact on the performance of two state-of-the-art models when applied to recognized text. Our experimental results and analysis show that extending word embeddings with pitch accent features slightly improves slot filling systems on recognized text.

1 Introduction

Slot filling is a subtask of spoken language understanding (SLU) that aims at assigning a semantic label to words in a sentence that “fill” a semantic frame, or slot, such as locations or time periods. State-of-the-art methods are evaluated on the benchmark dataset from the Airline Travel Information Systems (ATIS) corpus [1] and yield around 95% F1-score using deep neural network (DNN) architectures [2, 3, 4, 5, 6, 7]. The features used in these models are typically lexico-semantic representations in the form of word embeddings [8]. As slot filling operates on text only, typical experiments use the text data provided in the benchmark dataset along with the slot annotations. Since SLU is designed to extract information from speech and thus involves the use of automatic speech recognition (ASR), the more realistic setting would be to apply and optimize these tasks on actual ASR output. Mesnil et al. [5] report that the performance of their RNN-based slot filling model drops to around 85% F1-score on recognized text while that on the reference dataset is around 95%. He and Young [9] compare different SLU tasks on recognized and reference text and found that slot filling (referred to as semantic parsing) performance using a vector state model drops from around 90% to 89% F1-score.

For this reason, it may be helpful to include additional features that can be extracted from speech, such as prosodic information. Previous research has provided evidence that pitch accents are useful for various natural language processing tasks: Katerenchuck and Rosenberg [10] use prosodic labels in the form of ToBI types [11] and clusters of acoustic features to improve named entity recognition of recognized speech. Rösiger and Riester [12] found that knowing about the presence or absence of a pitch accent improves automatic coreference resolution, since coreferent items are given information in the discourse and hence typically deaccented. Several studies focused on the use of prosodic information for spoken language understanding, such as early experiments by Veilleux and Ostendorf [13] on the ATIS corpus and Shriberg and Stolcke [14] that used prosodic modelling to improve ASR and other SLU subtasks.

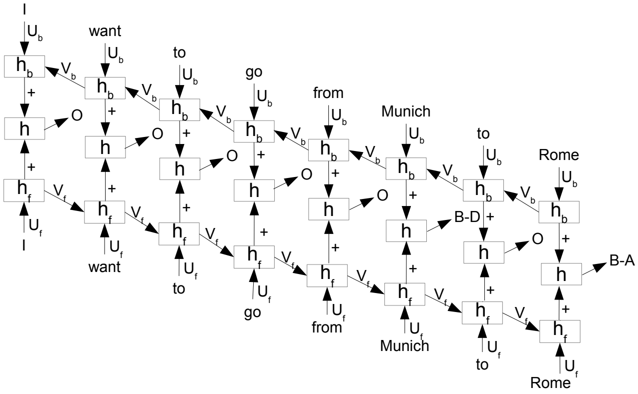

Figure 1 – Bidirectional recurrent neural network for the slot filling task

The intuition behind the use of prosodic information in this work can be explained as follows: During human-to-human discourse, there can be recognition errors due to, for example, environmental noise. A human listener may be able to fill a semantic slot not only by using the correctly understood context information, but also by prosodic cues. In this work we focus on pitch accents. Pitch accents are used to give words more salience during discourse to highlight certain information, such as focus, contrast or information status. For example, content words, especially with new information status, typically carry pitch accents [15]. An utterance in the ATIS dataset like *List flights from Dallas to Houston* is expected to have pitch accents on *Dallas* and *Houston*, since these will constitute new as well as key information in this setting. In a previous study [16] it was shown that words bearing automatically predicted pitch accents account for around 90% of the slots in a subset of the ATIS dataset. This observation holds when applied to ASR output, which means that such information can be extracted in an automated setting.

In this paper, we present a first simple, efficient step to extend two state-of-the-art DNN models for slot filling by adding prosodic information extracted automatically from speech data. This information is included as binary pitch accent features alongside the traditionally used lexico-semantic word embeddings. In this work we refer to these vectors as having *pitch accent extensions*. Previous research has shown that convolutional neural network (CNN) models based on high-dimensional word embeddings can benefit from even a few linguistically informed features [17]. We compare the effect this extension has on a bidirectional recurrent neural network (RNN) [6] and a bidirectional sequential CNN [7] applied to recognized text.

2 Pitch Accents in Neural Slot Filling Models

2.1 Neural Slot Filling Models

In order to build the baseline systems, we use the recurrent neural network (RNN) proposed by Vu et al. [6] and the convolutional neural network (CNN) proposed by Vu [7]. These two models utilize word embedding information in different ways. We examine the effect that adding pitch accent extensions to the word embeddings have on these models. The first model [6] is an elman-type bidirectional RNN (see Figure 1) that yields an F1-score of 95.56% on the benchmark ATIS dataset. The features in this model are 100-dimensional word embeddings that are randomly initizalized and jointly trained along with the RNN. The bidirectionality of this model refers to the combination of a forward and backward hidden layer using an addition operator in order to take past and future contexts (trigrams) into account.

The second model (see Figure 2) involves a bidirectional sequential CNN [7]. It outper-

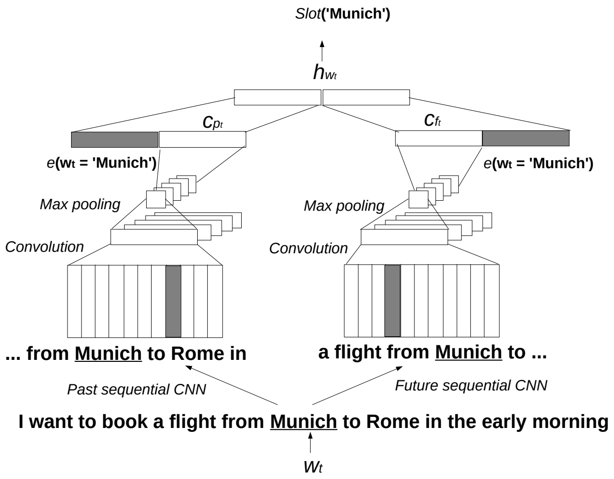

Figure 2 – Bidirectional sequential convolutional neural network for the slot filling task

forms the first model on slot filling task with a 95.61% F1-score on the ATIS dataset. This method combines two CNNs that model the future and past contexts respectively as well as an additional extended surrounding context to give the current word more weight. The presentation of the past and future contexts are concatenated to form the final automatically learnt features. The context windows for both the forward-moving and backward-moving CNNs comprise 9 words. The surrounding context hyperparameter is 3 words. The features are 50-dimensional word embeddings.

Objective Function Following Vu et al. [6] and Vu [7], we use the ranking loss function proposed by Santos et al. [18]. Instead of using the softmax activation function, we train a matrix W^{class} whose columns contain vector representations of the different classes. Therefore, the score for each class c can be computed by using the product

$$s_\theta(w_t)_c = h_{w_t}^T [W^{class}]_c \tag{1}$$

The ranking loss function [18] maximizes the distance between the true label y^+ and the best competitive label c^- given a data point x. The objective function is

$$L = \log(1 + \exp(\gamma(m^+ - s_\theta(w_t)_{y^+}))) + \log(1 + \exp(\gamma(m^- + s_\theta(w_t)_{c^-}))) \tag{2}$$

with $s_\theta(w_t)_{y^+}$ and $s_\theta(w_t)_{c^-}$ as the scores for the classes y^+ and c^- respectively. The parameter γ controls the penalization of the prediction errors and m^+ and m^- are margins for the correct and incorrect classes. γ, m^+ and m^- are hyperparameters which can be tuned on the development set. For the empty class O (no slot), only the second summand of Equation 2 is calculated during training, i.e. the model does not learn a pattern for class O but nevertheless increases its difference to the best competitive label. Furthermore, it implicitly solves the problem of unbalanced data since the number of class O data points is much larger than other classes. During testing, the model will predict class O if the scores for all other classes are < 0.

2.2 Extending the Word Embeddings with Pitch Accents

A word vector with a pitch accent extension for a word w consists of the word embedding of w as well as a binary flag $\in (0,1)$ indicating the absence or presence of a pitch accent on w:

$$embs(w) = [lexical\_embs(w), pitch\_accent\_flag(w)] \tag{3}$$

"1" refers to pitch-accented words and "0" to non-pitch-accented words. To extract this flag, we first run an ASR system on the audio signal to obtain the word, syllable, and phone alignments. Afterwards, a pitch accent detector (described in the next section) is applied to determine the binary label for each of the recognized words. The lexical word embeddings are randomly initialized and concatenated with the binary pitch accent flag to form the new vectors. Hence, words in different contexts might have different embeddings depending on whether they are pitch accented or rather not. This procedure constitutes a simple and efficient way to enhance word embeddings with pitch accent information for NLP tasks.

3 Pitch Accent Detection

The pitch accent detection model used in this work is trained on part of the Boston Radio News Corpus [19] that is labelled with prosodic events (ToBI accent and boundary types). This subset encompasses 220 speech files by 5 speakers, 3 female and 2 male, and consists of in total around 1 hour and 20 minutes of speech. We consider the binary case (pitch accent or none) and group all pitch accent types together as one class. The method, adopted from Schweitzer [20], requires time-aligned data in order to extract acoustic features for single syllables that are derived mostly from PaIntE parameters [21]. A further description of this pitch accent detection procedure is given in [16]. The cited work also shows that the accuracy of this model, when applied to the ATIS corpus, is roughly 70% when measured against a human labeller. This is only slightly lower than the accuracy measured using leave-one-speaker-out cross-validation on the Boston dataset, which is 74.4% on average.

4 Data

4.1 The ATIS Corpus

The ATIS corpus contains utterances of speakers requesting information on airline travel. Key query terms are assigned slot labels referring to semantic roles such as *departure date, departure time, airline name* while the rest are marked as empty, e.g. *I - O WANT - O TO - O FLY - O FROM - O DENVER - B-fromloc.city_name TO - O HOUSTON - B-toloc.city_name*.

The standard split used in He and Young [9] and subsequent related work consists of 4,978 utterances from the Class A training data of the ATIS-2 and ATIS-3 corpora. 893 utterances from ATIS-3 are used as test data. We use 4,924 utterances and corresponding .WAV files from the training dataset and use the 893 test utterances in this work. The reason for our smaller training set is the fact that the utterance IDs for the benchmark training dataset were not available to us and some sentences did not have a matching equivalent in the original corpus. This is necesssary, however, to create pairs of audio files and the respective transcriptions for our experiments.

4.2 Slot Labels for ASR Output

We recognized the ATIS test set using a triphone model from the Kaldi toolkit [1] trained on around 4,900 utterances from the ATIS training data with 7.06% word error rate (WER). In order to make the recognized output compatible with the slot filling system and comparable with the original test data, the following preprocessing steps were necessary.

Having obtained recognized text, we applied a workaround solution to the problem that the slot annotations were created for the original text, and we cannot simply transfer the annotations

[1] www.kaldi-asr.org

to the ASR output for evaluation since the text will differ slightly. We time-aligned the words in the original text to the audio files using Kaldi, which automatically provides the time intervals for each slot. Then, using the time-alignments of the recognized text, we assign each word in the recognized output a slot label according to the time intervals of the respective label and the reference word itself: If the reference and recognized words are the same, we assign the recognized word the label of the reference word if their time intervals match within a threshold of 0.05 seconds. Slot-annotated words in the reference that do not have an exact match in the recognized output are assigned to the recognized word within a time interval of 0.02 seconds. Using this method, we acquire annotations for the recognized output automatically, a form of *silver standard*, which we need in order to measure slot filling performance. Human-detectable inconsistencies can occur, however, due to recognition errors and and misaligned words, even though these remain at an acceptable level.

The slot filling training and test data from ATIS as used by Mesnil et al. [22] is provided as a python pickle object [2] that already contains the IDs for each word for use as feature vectors. We reused the same word-to-ID and label-to-ID indexes to create the new recognized test set.

5 Experimental Results

5.1 Pitch Accents in ATIS

In order to estimate how representative pitch accents are in the ATIS corpus, we count how often pitch accents co-occur with slots in the test set. We analyze the recognized version in the same manner. Table 1 shows the results of this analysis. Almost 93% of the words that are annotated with slot labels are also pitch accented. This holds for both the original transcriptions as well as the ASR output and provides evidence that was previously described in [16] that pitch accent features may serve as a resource for SLU tasks such as slot filling.

	original transcriptions	recognized text
# files	893	893
# words	9551	9629
# slots	3663	3560
# predicted accents	5295	5169
# pred. accents on slots	3395	3308
# pred. accents on non-slots	1900	1861
slots with pred. accent	92.7%	92.9%

Table 1 – Co-occurrences of pitch accents and slots in the original and recognized transcriptions of the ATIS test set.

5.2 Pitch Accents in Neural Models

As a preprocessing step before prosodic analysis, we time-align the full ATIS dataset at the phone, syllable and word level. We apply the pitch accent detector as described in section 3 to obtain the time points of every predicted pitch accent in the training data, the original transcriptions of the test data and the recognized test data. This new information is included in the input data to the slot filling models as a binary feature vector for each word: if there is an accent within the time interval of a word, then the feature value is 1, if not, then the value is 0. We add this information to the baseline models as described in section 2. The model is

[2] `http://deeplearning.net/tutorial/rnnslu.html`

trained using stochastic gradient descent with up to 100 epochs and early stopping. We begin training with an initial learning rate of 0.025 and halve the learning rate if the performance on the development set has not improved after 5 epochs. We compare the performance of the RNN and CNN slotfilling models described in section 2 on both the original text and the recognized output. The F1-scores are given in Table 2. Our results on the manual transcriptions are slightly different compared to the results reported in Vu et al. [6] and Vu [7] due to the fact that we used smaller training set and use 10% of the training data as a development set for early stopping.

	RNN	CNN
Transcriptions (lexical word embeddings)	94.97	95.25
+ pitch accent extensions	**94.98**	**95.25**
ASR output (lexical word embeddings)	89.55	89.13
+ pitch accent extensions	**90.04**	**89.57**

Table 2 – F1-scores of slot filling on original and recognized text and after adding pitch accent features.

As expected, results on the ASR output is much worse than on the manual transcriptions. This performance drop is due to ASR errors. The pitch accent extensions do not improve the F1-score when using manual transcriptions but do not harm the slot prediction performance either. This indicates that context information as modelled by the sequential model is strong enough to predict slot labels in this dataset. Adding pitch accent extensions, however, slightly improves the F1-score when on ASR output on both models (RNN and CNN). This implies that the added pitch accent information may help to balance out the effect of ASR errors in some cases.

In order to illustrate in what cases the pitch accent extensions can help improve the results, we present two example sentences in Table 3. These examples involve unknown tokens; tokens that replace words in the benchmark dataset that occur only once. The recognized version includes more unknown tokens in cases where the ASR system was not able to recognize a word, or when a misrecognized word does not exist in the original dataset. The unknown token serves as a "wildcard", receiving its own representation via the word embeddings but can be assigned any label. In the first sentence, the unknown token is assigned a location slot label by the model trained with pitch accent features while the baseline model assigned the empty slot. In the second sentence, the word *Toronto* has been misrecognized by the ASR system. The baseline model has trouble classifying the resulting words while the model using pitch accents "ignores" this part of the sentence, which would be considered correct in this case.

reference text	I NEED THE FLIGHTS FROM **WASHINGTON** TO MONTREAL ON A SATURDAY
recognized text	I NEED THE FLIGHTS FROM **<*UNK*>** TO MONTREAL ON SATURDAY
reference slots	O O O O O **B-fromloc.city_name** O B-toloc.city_name O B-depart_date.day_name
with accents	O O O O O **B-fromloc.city_name** O B-toloc.city_name O B-depart_date.day_name
baseline	O O O O O **O** O B-toloc.city_name O B-depart_date.day_name
reference text	WHICH AIRLINES FLY BETWEEN **TORONTO** AND SAN DIEGO
recognized text	WHICH AIRLINES FLY BETWEEN **TO ROUND <*UNK*>** AND SAN DIEGO
reference slots	O O O O **O O O** O B-toloc.city_name I-toloc.city_name
with accents	O O O O **O O O** O B-toloc.city_name I-toloc.city_name
baseline	O O O O **B-fromloc.city_name B-round_trip I-round_trip** O B-toloc.city_name I-toloc.city_name

Table 3 – Example utterances, recognized versions and their slot labels predicted by the RNN model with and without pitch accent extensions.

We ran a brief analysis of the RNN results to gain some insight into whether the pitch accent information helps to label slots on unkown words independent of the slot type. Specifically, we counted the instances where the model correctly distinguished slots from the empty class. Of all unknown words with a reference slot, around 43% were correctly labelled in the baseline model, whereas around 51% were correctly assigned a slot when using pitch accent extensions. This indicates that pitch accent information was used by the model to localize a semantic slot even though the predicted slot type was incorrect. Thus, unknown words may still originally carry acoustic information in the signal that is beneficial to this task, and is captured by the proposed method.

6 Conclusion

In this work, we addressed the notion of overcoming the performance drop of state-of-the-art slot filling methods on speech recognition output. Our method involved combining pitch accent features with word embeddings as a way of including acoustic-prosodic information that is extracted from the speech signal but is lost during ASR. We tested this method on the ATIS benchmark corpus using two different models. In terms of quantitative results, small but positive effects were obtained on both models. Taking a closer look reveals some evidence that pitch accent features may be helpful in the case of misrecognized or unknown words. While our analysis provides a straightforward intuition of the benefits of prosodic information for this task, it remains difficult to determine whether the proposed method or the fact that the ATIS dataset is rather limited is the reason why the performance increase is only slight. Further research is necessary to fully investigate the potential of prosodic information in slot filling.

References

[1] HEMPHILL, C. T., J. J. GODFREY, and G. R. DODDINGTON: *The ATIS Spoken Language Systems Pilot Corpus*. In *Proceedings of the DARPA Speech and Natural Language Workshop*. 1990.

[2] YAO, K., G. ZWEIG, M. HWANG, Y. SHI, and D. YU: *Recurrent neural networks for language understanding*. In *Proceedings of Interspeech*. 2013.

[3] XU, P. and R. SARIKAYA: *Convolutional neural network based triangular crf for joint intent detection and slot filling*. In *Proceedings of the IEEE ASRU Workshop*. 2013.

[4] YAO, K., B. PENG, Y. ZHANG, D. YU, G. ZWEIG, and Y. SHI: *Spoken language understanding using long short-term memory neural networks*. In *Proceedings of the IEEE Spoken Language Technologies Workshop*. 2014.

[5] MESNIL, G., Y. DAUPHIN, K. YAO, Y. BENGIO, L. DENG, D. HAKKANI-TUR, X. HE, L. HECK, G. TUR, D. YU, and G. ZWEIG: *Using recurrent neural networks for slot filling in spoken language understanding*. *IEEE Transactions on Audio, Speech and Language Processing*, 2015.

[6] VU, N. T., P. GUPTA, H. ADEL, and H. SCHÜTZE: *Bi-directional recurrent neural network with ranking loss for spoken language understanding*. In *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing*. 2016.

[7] VU, N. T.: *Sequential convolutional neural networks for slot filling in spoken language understanding*. In *Proceedings of Interspeech*. 2016.

[8] BENGIO, Y., R. DUCHARME, and P. VINCENT: *A neural probabilistic language model*. In *Proceedings of NIPS*. 2000.

[9] HE, Y. and S. YOUNG: *A data-driven spoken language understanding system.* In *Proceedings of the IEEE ASRU Workshop*. 2003.

[10] KATERENCHUCK, D. and A. ROSENBERG: *Improving named entity recognition with prosodic features.* In *Proceedings of Interspeech*. 2014.

[11] SILVERMAN, K., M. BECKMAN, J. PITRELLI, M. OSTENDORF, C. WIGHTMAN, P. PRICE, J. PIERREHUMBERT, and J. HIRSCHBERG: *Tobi: A standard for labeling english prosody.* In *Proceedings of the International Conference on Spoken Language Processing*. 1992.

[12] RÖSIGER, I. and A. RIESTER: *Using prosodic annotations to improve coreference resolution of spoken text.* In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. 2015.

[13] VEILLEUX, N. M. and M. OSTENDORF: *Prosody/parse scoring and its application in ATIS.* In *In Proceedings of the ARPA Workshop on Human Language Technology*. 1993.

[14] SHRIBERG, E. and A. STOLCKE: *Prosody modeling for automatic speech recognition and understanding.* In *Mathematical Foundations of Speech and Language Processing*. 2004.

[15] HIRSCHBERG, J. and J. PIERREHUMBERT: *The intonational structuring of discourse.* In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*. 1986.

[16] STEHWIEN, S. and N. T. VU: *Exploring the correlation of pitch accents and semantic slots for spoken language understanding.* In *Proceedings of Interspeech*. 2016.

[17] EBERT, S., N. T. VU, and H. SCHÜTZE: *A linguistically informed convolutional neural network.* In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 2015.

[18] SANTOS, C. N. D., B. XIANG, and B. ZHOU: *Classifying relations by ranking with convolutional neural networks.* In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. 2015.

[19] OSTENDORF, M., P. PRICE, and S. SHATTUCK-HUFNAGEL: *The Boston University Radio News Corpus.* Technical Report ECS-95-001, 1995.

[20] SCHWEITZER, A.: *Production and Perception of Prosodic Events-Evidence from Corpus-based Experiments.* Ph.D. thesis, Universität Stuttgart, 2010.

[21] MÖHLER, G. and A. CONKIE: *Parametric modeling of intonation using vector quantization.* In *Proceedings of the 3rd ESCA Workshop on Speech Synthesis*. 1998.

[22] MESNIL, G., X. HE, L. DENG, and Y. BENGIO: *Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding.* In *Proceedings of Interspeech*. 2013.