IDIAP RESEARCH REPORT

STACKED NEURAL NETWORKS WITH PARAMETER SHARING FOR MULTILINGUAL LANGUAGE MODELING

Banriskhem Khonglah Srikanth Madikeri
Navid Rekabsaz Nikolaos Pappas Petr Motlicek
Hervé Bourlard

Idiap-RR-12-2019

OCTOBER 2019

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Stacked Neural Networks with Parameter Sharing for Multilingual Language Modeling

Banriskhem K. Khonglah, Srikanth Madikeri, Navid Rekabsaz, Nikolaos Pappas, Petr Motlicek, Hervé Bourlard

Idiap Research Institute, Martigny, Switzerland

{banriskhem.khonglah, srikanth.madikeri, navid.rekabsaz, nikolaos.pappas, petr.motlicek, herve.bourlard}@idiap.ch

Abstract

Neural language models (NLM) are an important component of Automatic Speech Recognition (ASR), providing effective re-scoring capability. Neural multilingual models have gained significant attraction by transferring knowledge across languages, especially to the ones with limited domain-specific data. Despite several studies on learning multilingual acoustic models, there is lack of understanding of the effects of novel multilingual training mechanisms for language modeling. In this work, we propose a neural model, consisting of multiple language-specific layers and one language-independent layer, trained in a multilingual setting. Our proposed model consists of time delay neural network components for the language-specific layers, and long short term memory for the language-independent layer. The former captures the characteristics of the individual languages, and the latter learns the common sentence structures. We evaluate our model on four BABEL languages in terms of perplexity for language modeling and Word Error Rate (WER) for ASR. Evaluation results demonstrate the strengths of our multilingual neural model. On Tagalog and Swahili, our model improves over previous monolingual and multilingual baselines in both perplexity and WER, while on Turkish and Zulu, which are high inflectional languages, it is not far behind N-gram models which exhibit better performance than the neural-based models.

Index Terms:Multilingual language modeling, neural networks, Re-scoring for speech recognition

1. Introduction

Neural Network Language Models (NNLMs) are widely used for re-scoring word recognition hypotheses in Automatic Speech Recognition (ASR) systems. There are applications where the specific domain of data needs to be improved for ASR. One such application is in the MATERIAL program [1]. In this program, the ASR for low-resource languages is attempted for document retrieval and summarization purposes. There are different domains of data in this program, like broadcast news and conversational speech. The availability of data related to broadcast news is abundant in nature (can be mined from the web) and hence these domains can be improved easily for ASR by adding more data to the language model. However, for conversational speech, there is less availability of data. For such domains, multilingual NNLMs sharing parameters across multiple languages can be used. These models aim to address these data sparsity issues [1]. In general, such multilingual models have language-specific as well as language-independent parameters (i.e. shared among all languages). Parameter sharing across multiple languages may act as implicit regularization for the NNLMs involved, especially for languages with insufficient amount of domain-specific data, due to the knowledge transfer that is occurring across languages [2].

Multilingual training of neural networks has grown in the last few years on acoustic modeling for ASR [3–5], as well as for language processing tasks such as document classification [2] and machine translation [6]. Ragni et al. [1] proposed a multilingual neural language model, inspired from previous work on multilingual acoustic modeling in ASR [4]. Their model comprises of a Recurrent Neural Network (RNN) with one layer where the weights of the hidden layer of the RNN are shared across the languages, but the input and output layers are language-specific. Furthermore, the multilingual model in [1] is further fine-tuned on each language. Lastly, the input and output layers in [1] contain the vocabulary lists of all languages, meaning that the loss is computed over the entire vocabularies of languages.

In the aforementioned work, the approach followed can cause bias when computing Softmax in the output layer for a given language, since unrelated vocabularies (the ones in other languages) are considered in the normalization factor of Softmax for the vocabularies of that language. In this work, we pursue the direction of [1], by proposing a state-of-the-art approach for multilingual neural language modeling applied in re-scoring hypotheses in ASR. Our proposed multilingual architecture consists of a stacked neural network model, where the first layer is language-specific and the second one is shared across multiple languages. In addition, in contrast to [1], every language has separate input and output layers and hence a separate loss function. The overall loss in our proposed approach is the weighted sum of per-language loss values, used to optimize the whole network through back-propagation (details in Section 3).

Various forms of RNN models have been used for language modeling [7–10], among which, Long Short Term Memory (LSTM) networks demonstrate the best and reliable performance [11]. However, recent work has shown that the infinite memory capacity of RNN may be actually absent in practice [12–14]. Nevertheless, the RNN has the capability to capture the common sentence structures among various languages [15]. The feed-forward neural networks, especially the Time Delay Neural Network (TDNN) models, have also shown competitive performance compared to LSTMs, if sufficient context is provided. In the light of these studies, we explore the combination of these two architectures (TDNN and LSTM) for multilingual language modeling. The TDNN will be used for capturing the characteristics of the individual languages and hence acts as the language-specific layer. The LSTM will be

[1] https://www.iarpa.gov/index.php/research-programs/material

used for capturing the common sentence structures and represents the language-independent layer. The specific contributions of the paper are the following:

- Exploring a joint multilingual objective function for language modeling based on the weighted sum of the per-language losses.
- Designing a TDNN-LSTM stacked architecture where the TDNN captures the characteristics of the individual languages while the LSTM captures the common sentence structures.

Our experimental results show noticeable improvement of the proposed multilingual models in comparison to monolingual models as well as previously proposed multilingual approaches for two (Tagalog and Swahili) out of the four languages of interest, both in terms of perplexity for language modeling and Word Error Rate (WER) for ASR systems. There is no improvement on the other two languages (Zulu and Turkish) probably due to the fact that they are high inflectional languages with more out of vocabulary words. Simpler N-gram model with lesser parameters provides better performance for these languages. Even though the improvements brought by our multilingual model are not observed in every language used in our experiments, the WER is always at least as good or better than the performance based on monolingual models. Furthermore, we also observe a link between improvement in perplexity and WER for Tagalog and Swahili which is a promising finding.

Our TDNN-LSTM architecture is also compared to the other combinations like TDNN-TDNN or LSTM-LSTM to show that for capturing the particular characteristics of individual languages, the TDNN outperforms other architectures and for capturing the common sentence structures across languages, the LSTM is better. We show that our proposed method of combining the networks (assuming the training principle remains the same) enables a complementary capture of information by the two networks for multilingual language modeling. As opposed to [1], our proposed model does not require fine-tuning for obtaining gains in performance, saving the re-training phase on each language, but could perhaps benefit from it separately.

The paper is organized as follows. Section 2 provides background on language models in ASR as well as the multilingual acoustic model. The proposed multilingual language model is described in Section 3. The results are presented and discussed in Section 4, and Section 5 concludes the study.

2. Background

2.1. Language Modeling in ASR

Language modeling in ASR is typically based on the N-grams which involve estimating the probability based on counting. Later smoothing techniques were introduced to make the N-grams more robust to zero counts [16]. Given the recent advances in language modeling using neural networks, one way of exploiting them in ASR is by re-scoring the N-best list of word recognition hypotheses obtained from the decoding based on N-grams [7]. This is usually called the second pass decoding. There are also attempts to use the neural networks for first pass decoding. In [17], the RNN is used as a generative model to generate text. The N-gram LMs are then trained based on the generated text and finally used for decoding. In [18], RNN LM histories are discretized to create Weighted Finite State Transducers (WFST). A probability-based conversion has been explored in [19], and this method involves the extraction of N-grams but the count based probabilities are replaced by the RNN-LM probabilities. In the present study, we use the second pass decoding, where the neural language model is used to re-score the N-best hypothesis generated by the N-gram language model.

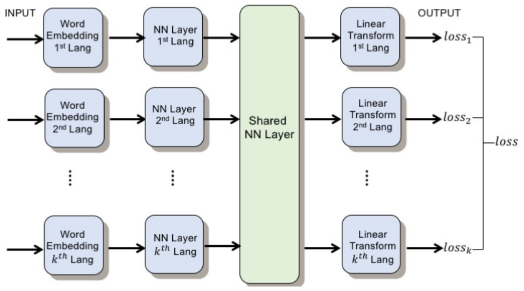

Figure 1: *Schematic of the proposed multilingual neural language model architecture with stacked layers. The elements in blue are language-specific and the green ones are language-independent. NN indicates a neural network model which is either TDNN or LSTM in our study.*

2.2. Multilingual Acoustic Modeling

To train acoustic models for ASR with limited amount of data, it is helpful to augment data to avoid sub-optimal convergence. There are multiple ways for data augmentation. One method is to create multiple copies of the training data by adding noise or varying speed and volume. Another efficient method is to train a multi-task network. In a multi-task network, it is possible to combine several low-resource languages to eliminate the data insufficiency problem. In [20], 19 languages from the Babel program are used to train a BLSTM based acoustic model. We consider a similar approach in this paper, where a single multilingual TDNN acoustic model is trained for all the four languages. A 5-layer TDNN is trained with a block-softmax output, with one output block for each of the languages. During training of the network, each mini-batch is balanced to contain examples from all languages. We use parallel training proposed in [21] for this purpose. The initial alignments for training the multilingual model are obtained from monolingual HMM/GMM acoustic models.

3. Multilingual Language Model

In this section, we describe the architecture of our proposed multilingual neural language model, depicted in Figure 1, and explain its training procedure.

As shown in Figure 1, for a given language l, first a language-specific word embedding maps the input batch of the given language to its embedding vectors. A language dependent neural network layer (TDNN) then captures the language-specific characteristics of the input, followed by a language-independent neural network layer (LSTM), where its parameters are shared across all languages. In the next step, a language-specific linear transform, followed by the softmax function, provides the predicted probability distribution $\hat{y}_t^{(l)}$ at timestep t over the vocabulary of the given language l.

For training, we use a joint multilingual objective function that facilitates the sharing of a subset of parameters for each language $\theta_1, \ldots, \theta_M$ of our stacked neural language network

Table 1: *Statistics of the four BABEL languages.*

Statistics ↓	**Languages**			
	Tagalog	**Swahili**	**Turkish**	**Zulu**
no. of sentences (train)	93131	39354	82253	54660
no. of words (train)	594854	250398	573323	369476
no. of sentences (dev)	11191	9678	10297	9163
no. of words (dev)	73143	62875	73306	58285
vocab. size	22907	23956	41196	56885

as in [2]:

$$\mathcal{L}(\theta_1, \ldots, \theta_M) = -\frac{1}{Z} \sum_{t}^{N_e} \gamma_l \sum_{l}^{M} \mathcal{H}(y_t^{(l)}, \hat{y}_t^{(l)}) \qquad (1)$$

where $Z = M \times N_e$, N_e is the epoch size, M is the total number of languages, γ_l is a hyper-parameter for each language objective which encodes prior knowledge about its importance and $\mathcal{H}$ is the cross-entropy loss between the ground-truth words and the predicted ones. Note that the sentence order in each language is preserved above and that the overall loss is back-propagated through the network, updating both language-specific and language-independent parameters. The sentences are processed in a cyclic fashion for the languages which have lesser number of sentences. Once the last sentence of the text corpus is processed for that language, the next sentence that is processed is the beginning one. The joint objective $\mathcal{L}$ can be minimized with respect to the parameters $\theta_1, \ldots, \theta_M$ using Stochastic Gradient Descent (SGD). This training strategy has been shown beneficial in the past for multilingual document classification [2] and multilingual neural machine translation [22].

4. Experiments and Results

The experiments are performed on four babel languages, released as a part of the IARPA Babel program, namely Tagalog, Swahili, Turkish and Zulu. The detailed statistics of each language set are mentioned in Table 1. Each language has a held out development set, used for reporting the perplexity of language models as well as WER of the ASR systems. SRILM toolkit [23] is used to create N-gram language models. The neural language models are implemented in *pytorch*. A modified version of TDNN, released in [12] is used in our experiments. We also implement a multilingual neural language model with RNN, following the work in [1]. The acoustic models are trained using the Kaldi speech recognition toolkit [24].

The neural networks are optimized using the Stochastic Gradient Descent algorithm with early stopping, and negative log likelihood as the loss function. One-hot encoding is used for the input layers with the size of the number of vocabularies in each language. The dimensions of the hidden nodes of TDNN, LSTM, and RNN as well as the word embedding are set to 600. The feed-forward architecture of the TDNN nodes consist of three hidden layers. All hyper-parameters of the model are set according to the best results, evaluated on the development set. The parameter γ_l in Equation 1 is not explored exhaustively for the range of values but set at the value of $1/M$, where M is the total number of languages.

Table 2: *Perplexity on the Babel Set of four languages for monolingual and multilingual LMs are presented. For the stacked models, the first network is language-dependent and the second one is shared. Full: full vocab size and 20K:vocab size of 20,000. Tag: Tagalog, Swa: Swahili, Tur: Turkish, Zul: Zulu*

LM ↓	**Perplexity**							
	Full				**20K**			
	Tag	**Swa**	**Tur**	**Zul**	**Tag**	**Swa**	**Tur**	**Zul**
N-gram	148	357	396	719	109	155	183	131
RNN (multi) [1]	142	294	**284**	**665**	104	218	**117**	115
TDNN-LSTM (mono)	136	383	422	725	130	239	168	124
LSTM-LSTM (multi)	159	302	509	1550	262	237	351	122
TDNN-TDNN (multi)	137	622	293	1485	**102**	184	148	186
TDNN-LSTM (multi)	**133**	**284**	337	1006	107	**147**	150	**108**

4.1. Language Modeling Results

As a baseline, we first compute the perplexity using N-gram LMs. The N-gram used in this work is a tri-gram. We also experimented with a four-gram but it was performing worse than the tri-gram. The perplexities using the TDNN-LSTM are shown in the last row of Table 2. For comparison, other architectures are also considered in Table 2. Initially if the entire vocabulary is used for each language, the perplexity is recorded and shown as the ”Full” column of Table 2. It can be seen that using the entire vocabulary results in larger perplexities for Zulu, Turkish and Swahili. For Tagalog, the perplexities are comparatively better.

The reason why Zulu, Turkish and Swahili have large perplexities can be understood from Table 1. Consider the ratio of the training data to the vocabulary size for each language. It can be seen that this ratio is the highest for Tagalog which means there is enough training data for training the LM with the respective vocabulary for Tagalog. For Swahili, even though the vocabulary size is almost similar to Tagalog, the amount of training data is much less. Hence this results in larger perplexities. Zulu has the largest vocabulary which is almost double of Swahili and has the training data much less than Tagalog although slightly more than Swahili. This results in the very large perplexities for Zulu. Turkish has almost the same amount of data as Tagalog, however, it has a vocabulary which is almost twice as big compared to Tagalog. This causes Turkish to have higher perplexities. These perplexities computed on the entire vocabulary are noisy and this problem is usually solved by limiting the vocabulary size as done in [1].

In [1], the vocabulary size for each language was limited to 75% of the total vocabulary size, with the remaining words mapped to an unknown symbol ”unk”. In this work, a similar approach is followed. However the vocabulary size of each language is limited to ”20K”. This value is chosen taking into account the lowest vocabulary size of the four languages. The remaining words out of the ”20K” are mapped to ”unk”. By doing this, the perplexities of each language are more stable as seen in Table 2 for the column marked as ”20K”.

The multilingual TDNN-LSTM LM is found to perform well in terms of perplexity compared to the N-grams and also outperforms the monolingual TDNN-LSTM LM for the

”Full” vocabulary case. Improvements can be seen for Tagalog, Swahili and Turkish although degradation is observed for Zulu. Similar observations are found for the ”20K” vocabulary as well, although for this case, even for Zulu the performance is better. The multilingual TDNN-LSTM LM is also compared to the TDNN-TDNN and LSTM-LSTM LMs. The LSTM-LSTM LMs suffers degradation in perplexity compared to the TDNN-LSTM LM as seen in table 2 for both the ”Full” and the ”20K” vocabularies. The TDNN-LSTM LM also outperforms the TDNN-TDNN LM except for Turkish in the ”Full” vocabulary case. Using the ”20K” vocabulary, the TDNN-LSTM LM is better than the TDNN-TDNN LM for Swahili and Zulu, although it is slightly worse for Tagalog and Turkish. The results using the RNN LM proposed in [1] is also computed and the TDNN-LSTM system is better except for Zulu and Turkish using the ”Full” vocabulary while using the ”20K” vocabulary, the results using TDNN-LSTM LM are better for Swahili and Zulu. It is comparable for Tagalog but worse for Turkish. Overall, the TDNN-LSTM LMs display good perplexity behavior in at least two of the four languages and results indicate that they perform well on languages with a small vocabulary. This can be further validated by the fact that reducing the vocabulary of Zulu helps in improving the perplexity.

In the following section, the proposed models are tested in terms of the word error rate for ASR.

4.2. ASR Results

The perplexities of the four languages were seen to be more stable using the reduced vocabulary. Similarly for the ASR task, the vocabulary of each language is limited as above. The N-gram LMs are initially used for creating a graph for ASR decoding. Multilingual acoustic models are used and the setup has been described in Section 2.2. The results using the N-gram decoding are shown in Table 3. The proposed neural networks are then used to re-score the N-best (N=1000) word recognition hypothesis generated from the lattices constructed using N-gram with weights (0.75 for the neural network and 0.25 for the N-gram) [25].

It can be seen that the Multilingual TDNN-LSTM outperforms the N-gram in terms of WER and also better than the monolingual TDNN-LSTM system on Tagalog and Swahili. Comparisons are made to the other multilingual LMs and it can be seen that the TDNN-LSTM LM is better. The TDNN-LSTM multilingual system does not perform better for Zulu and Turkish. Zulu and Turkish are languages with high inflections. On reducing the vocabulary, the number of out-of-vocabulary words in these languages increases which hurts the ASR performance. At the same time if the full vocabulary is used for these languages, the perplexity increases significantly as seen in Table 2 which will further hurt the ASR performance. This is due to the larger number of parameters that will be required and hence even more data is required for training. Simpler models like N-grams with less parameters appear to be more effective for these languages. Overall, the WERs for Swahili are better than the ones reported in [1] even without applying the fine-tuning. The relative improvement in WER for Tagalog is 2 % and for Swahili is 1 % with respect to the N-grams.

Table 3: *WER on the Babel Set of four languages for N-gram, monolingual and multilingual LMs. Our model improves significantly WER on two out of four languages without hurting significantly the performance of the other two.*

LM ↓	**WER %**			
	Tagalog	**Swahili**	**Turkish**	**Zulu**
n-gram (baseline)	44.5	35.4	**46.1**	**54.2**
RNN (multi) [1]	44.7	35.3	46.3	55.4
TDNN-LSTM (mono)	43.8	35.3	46.7	55.0
LSTM-LSTM (multi)	44.8	35.7	47.2	55.9
TDNN-TDNN (multi)	44.4	35.5	46.3	55.8
TDNN-LSTM (multi)	**43.6**	**35.0**	46.5	55.3

5. Conclusion

This work examines the use of multilingual language models using neural architectures. A stacked TDNN-LSTM architecture is used where the TDNN models the long context and the LSTM models the sentence structure. Training the multilingual LMs involves adding the losses of each language and the total loss is back-propagated. Experiments show that the multilingual TDNN-LSTM architecture outperforms N-grams and other stacked neural architectures on two out of four languages in terms of both perplexity and word error rate. In the future, more languages will be used for training, while adaptation to a particular unseen language can also be performed as done in [1], to further improve the perplexity and word error rate. In this work, the weights of the multilingual LMs have not been explored exhaustively for the range of values but set at the equal values. In the future, the weights can be tuned more effectively to get better performances.

6. Acknowledgement

The research is based upon the work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract #FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

7. References

[1] A. Ragni, E. Dakin, X. Chen, M. J. Gales, and K. M. Knill, “Multi-language neural network language models.” in *Interspeech*, 2016, pp. 3042–3046.

[2] N. Pappas and A. Popescu-Belis, “Multilingual hierarchical attention networks for document classification,” in *Proc. IJNLP*, 2017.

[3] S. Tong, P. N. Garner, and H. Bourlard, “An investigation of deep neural networks for multilingual speech recognition training and adaptation,” in *Interspeech*, 2017.

[4] K. Veselỳ, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, “The language-independent bottleneck features,” in *SLT Workshop*. IEEE, 2012, pp. 336–341.

[5] A. Ghoshal, P. Swietojanski, and S. Renals, “Multilingual training of deep neural networks,” in *ICASSP*. IEEE, 2013, pp. 7319–7323.

[6] M. Johnson *et al.*, “Google’s multilingual neural machine translation system: Enabling zero-shot translation,” *Transactions of the Association of Computational Linguistics*, 2017.

[7] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Interspeech*, 2010.

[8] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," in *Interspeech*, 2011.

[9] X. Liu, Y. Wang, X. Chen, M. J. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *ICASSP*. IEEE, 2014, pp. 4908–4912.

[10] H. Xu, K. Li, Y. Wang, J. Wang, S. Kang, X. Chen, D. Povey, and S. Khudanpur, "Neural network language modeling with letter-based features and importance sampling," in *ICASSP*. IEEE, 2018.

[11] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," in *Proc. ICLR*, 2018.

[12] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[13] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[14] J. Miller and M. Hardt, "When recurrent models don't need to be recurrent," *arXiv preprint arXiv:1805.10369*, 2018.

[15] T. Wada and T. Iwata, "Unsupervised cross-lingual word embedding by multilingual neural language models," *arXiv preprint arXiv:1809.02306*, 2018.

[16] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.

[17] A. Deoras, T. Mikolov, S. Kombrink, M. Karafiát, and S. Khudanpur, "Variational approximation of long-span language models for lvcsr," in *ICASSP*. IEEE, 2011, pp. 5532–5535.

[18] G. Lecorvé and P. Motlicek, "Conversion of recurrent neural network language models to weighted finite state transducers for automatic speech recognition," in *Interspeech*, 2012.

[19] H. Adel, K. Kirchhoff, N. T. Vu, D. Telaar, and T. Schultz, "Comparing approaches to convert recurrent neural networks into backoff language models for efficient decoding," in *Interspeech*, 2014.

[20] M. Karafiát, M. K. Baskar, P. Matějka, K. Veselỳ, F. Grézl, and J. Černocky, "Multilingual blstm and speaker-specific vector adaptation in 2016 but babel system," in *SLT Workshop*. IEEE, 2016, pp. 637–643.

[21] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[22] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proc. NAACL-HLT*, San Diego, CA, USA, June 2016, pp. 866–875.

[23] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh international conference on spoken language processing*, 2002.

[24] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[25] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "Rnnlm-recurrent neural network language modeling toolkit," in *Proc. of the 2011 ASRU Workshop*, 2011, pp. 196–201.