Joint-Character-POC N-gram Language Modeling For Chinese Speech Recognition

Bin Wang[1], *Zhijian Ou*[1], *Jian Li*[2], *Akinori Kawamura*[3]

[1]Department of Electronic Engineering, Tsinghua University
[2]Toshiba (China) Co., Ltd.
[3]Corporate Research and Development Center, Toshiba Corporation
Corresponding email: ozj@tsinghua.edu.cn

Abstract

The state-of-the-art language models (LMs) for Chinese speech recognition are word n-gram models. However, in Chinese, characters are morphological in meaning and words are not consistently defined. There are recent interests in building the character n-gram LM and its combination with the word n-gram LM. In this paper, in order to exploit both character-level and word-level constraints, we propose the joint n-gram LM, which is an n-gram model based on joint-state that is a pair of character and its position-of-character (POC) tag. We point out the pitfall in naive solving of the smoothing and scoring problems for joint n-gram models, and provide corrected solutions. For experimental comparison, different LMs (including word 4-grams, character 6-grams and joint 6-grams) are tested for speech recognition, using training corpus of 1.9 billion characters. The joint n-gram LM achieves performance improvements, especially in recognizing the utterances containing OOV words.

Index Terms: Chinese Speech Recognition, Language Model, Joint n-gram

1. Introduction

Language modeling plays an important role for speech recognition. It is essentially to estimate the distribution over all possible sentences in the target language. The state-of-the-art language models (LMs) are word-based n-gram models. However, the concept of word in Chinese is rather vague [1]. There are no delimiters between adjacent Chinese words in a sentence, and there is even no standard definition of a word in Chinese. Moreover, it is always possible to construct new words by combining multiple characters, which causes the out-of-vocabulary (OOV) problem. Considering these characteristics of Chinese language, there are recent interests in building character-based Chinese LMs, which typically are character n-grams [2, 3, 4]. The advantage is that it eliminates the OOV problem and avoids the complication of using an arbitrary lexicon. It is found in [3] that using the character-based LM alone produces slightly worse error rates than using the word-based LM alone, and combining the two gives better results than either model separately. This is presumably because that the constraints imposed by character-based n-grams are not as restrictive as those imposed by word-based n-grams, and the two types of constraints complement to each other. In this paper, we explore an alternative approach to exploiting both types of linguistic constraints.

This approach is inspired by the recent great success of using conditional random fields (CRFs) for Chinese word segmentation (CWS) [5, 6], which introduces the position-of-character (POC) tags. The POC tag of a character could take four possible values - B, M, E and S, which represents the beginning, middle, end of a word and a single-character word respectively. The CWS problem is thus solved as character-sequence tagging, with the ability of recalling OOV words. However, the CRFs used in the CWS studies are in essence not language models and cannot be used for speech recognition. They are not generative models $p(x)$ of Chinese sentences x, but conditional models $p(y|x)$ of POC tag-sequence y given character-sequence x.

Motivated by the above observations, we propose to augment character-based LMs with POC tags which carry word-level constraints. In particular, we pair every character c_i in a sentence with its POC tag g_i, which defines a joint-state $[c_i, g_i]$. We then model the joint-state sequence as a Markov source of order $n-1$, which we call a joint-character-POC n-gram LM (abbreviated as joint n-gram LM). It is a truly generative model of Chinese sentences which could be used in speech recognition, and has the potential for modeling both character-level and word-level linguistic constraints. For experimental comparison, three types of LMs (word 4-gram, character 6-gram and joint 6-gram) are tested for speech recognition, using training corpus of 1.9 billion characters. Compared to both the word 4-gram and the character 6-gram LM, the joint 6-gram LM achieves better performance, especially in recognizing the utterances containing OOV words. Moreover, we examine the combination of joint 6-gram with word 4-gram. Compared to the combination of character 6-gram with word 4-gram, the new combination still shows the advantage in handling OOV words.

The rest of this paper is organized as follows. Section 2 introduces the new joint n-gram LM. After the model definition, we focus on two basic problems when applying the new LM in speech recognition - the smoothing problem and the scoring problem. We point out some pitfalls in naive solving of the two problems for joint n-gram models, and provide corrected solutions. Section 3 describes the results from our experiments, which demonstrate the effectiveness of joint n-gram LMs. Finally, in Section 4, we discuss related work and point out future

POC tag g_{i-1}	Following legal POC tags g_i
B	M / E
M	M / E
E	B / S
S	B / S

Table 1: Hard constraints between adjacent POC tags, g_{i-1} and g_i

work.

2. Joint n-gram language models

2.1. Model definition

For a sentence represented as a sequence of linguistic units $u_1, u_2, \ldots, u_L$, an n-gram LM is defined as:

$$p(u_1^L) \triangleq \prod_{i=1}^{L} p(u_i|u_{i-n+1}^{i-1}) \tag{1}$$

where u_i is the linguistic unit at position i, L is the length of the sentence in terms of such units, and u_i^j denotes the units $u_i, \ldots, u_j$. The conventional choice for the linguistic units in Chinese could be words or characters, which leads to word-based n-gram LMs and character-based n-gram LMs respectively. In contrast, a joint-character-POC n-gram LM (abbreviated as joint n-gram LM) is an n-gram model based on joint-states, by defining $u_i \triangleq [c_i, g_i]$, where c_i is the character and and g_i is the POC tag of c_i. The POC tag of a character could take four possible values - B, M, E and S, which represents the beginning, middle, end of a word and a single-character word respectively. The word-based n-gram LMs in Chinese suffer from the OOV problem and the complication of using an arbitrary lexicon. The character-based n-gram LMs suffer from the loss of word-level constraints which could be incorporated with the help of POC tags. By this analysis, joint n-gram models appear to be a better choice for Chinese LMs.

Given the above form of the joint n-gram model, there are two basic problems that must be solved for the new model to be useful in speech recognition, namely the smoothing problem and the scoring problem.

2.2. Smoothing

The smoothing procedure is used to avoid the overfitting of the maximum likelihood (ML) estimation of probabilities. A large number of smoothing methods for n-gram models have been studied and compared in [7]. At first thought, we could just apply any of the existing smoothing methods (e.g. the well-known modified Kneser-Ney smoothing method) to the joint n-gram models. But as we explain in the following, there is a pitfall in such straightforward application.

Most existing smoothing algorithms for n-gram models can be described as follows:

$$p(u_i|u_{i-n+1}^{i-1}) = \begin{cases} \alpha(u_i|u_{i-n+1}^{i-1}) & \text{if } C(u_{i-n+1}^{i}) > 0 \\ \gamma(u_{i-n+1}^{i-1})p(u_i|u_{i-n+2}^{i-1}) & \text{if } C(u_{i-n+1}^{i}) = 0 \end{cases} \tag{2}$$

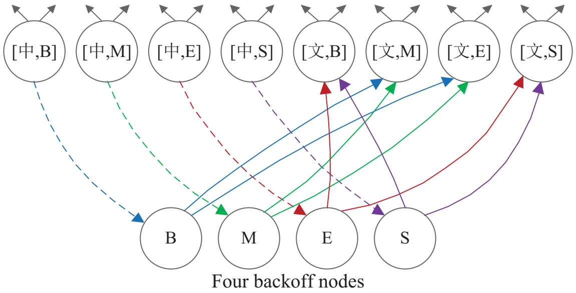

Figure 1: A fragment of the revised WFST representation for an example joint n-gram LM

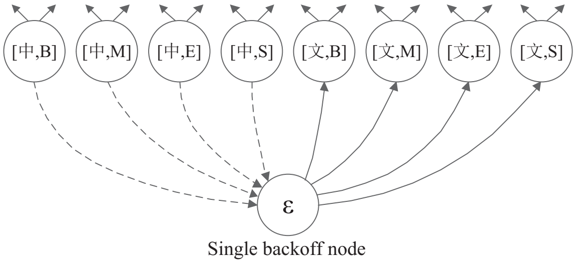

Figure 2: A fragment of the 'standard' (but false) WFST representation for an example joint n-gram LM

where $C(u_i^j)$ denotes the number of times u_i^j occurs in the training data. That is, if an n-gram u_{i-n+1}^{i} occurs in the training data, the estimate $\alpha(u_i|u_{i-n+1}^{i-1})$ is used, which is generally a discounted version of the ML estimate. Otherwise, we back off to a scaled version of the (n-1)-gram distribution $p(u_i|u_{i-n+2}^{i-1})$. The lower-order distribution $p(u_i|u_{i-n+2}^{i-1})$ is defined analogously to the higher-order distribution, and the recursion usually ends with the unigram distribution. The scaling factor $\gamma(u_{i-n+1}^{i-1})$ is chosen to assure that each conditional distribution sums to one.

The key observation of the pitfall is that there exist hard constraints between adjacent joint-states, $u_{i-1} \triangleq [c_{i-1}, g_{i-1}]$ and $u_i \triangleq [c_i, g_i]$. Specifically, it is the hard constraints between the adjacent POC tags, g_{i-1} and g_i. As shown in Table 1, each of the four POC tags can be followed by only two out of the four tags. Therefore, if we simply apply Equ.(2) to joint n-grams, some probabilities are improperly assigned to impossible transitions between joint-states due to smoothing.

To make corrections, an important observation is that if smoothed bigram estimates $p(u_i|u_{i-1})$ are corrected, then higher-order probabilities $p(u_i|u_{i-n+1}^{i-1})$ $(n \geq 3)$ could be correctly estimated just by applying Equ.(2) recursively. The revised formula for estimating the joint bigram probabilities is as follows:

$$p([c_i, g_i]|[c_{i-1}, g_{i-1}]) = \begin{cases} \alpha([c_i, g_i]|[c_{i-1}, g_{i-1}]) & \text{if } C([c, g]_{i-1}^{i}) > 0 \\ \gamma([c_{i-1}, g_{i-1}])p([c_i, g_i]) & \text{if } C([c, g]_{i-1}^{i}) = 0 \\ & \quad \text{and } g_{i-1} \rightarrow g_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $g_{i-1} \rightarrow g_i$ denotes that the POC connection $g_{i-1}g_i$ is legal. Again, the scaling factor $\gamma([c_{i-1}, g_{i-1}])$ is recalculated to assure that each conditional distribution sums to one.

2.3. Scoring

Scoring is needed for perplexity calculation, and more importantly, for N-best rescoring or lattice rescoring to evaluate different LMs in speech recognition. The scoring problem is how we compute the probability that a Chinese sentence (unsegmented, namely a sequence of Chinese characters) is produced by a given joint n-gram LM. Theoretically, this probability, often referred to as the LM score, is computed as follows:

$$p(c_1^L) = \sum_{g_1^L} p([c,g]_1^L) = \sum_{g_1^L} \prod_{i=1}^{L} p([c_i, g_i] | [c,g]_{i-n+1}^{i-1}) \quad (4)$$

In practice, Viterbi approximation is often used to max-marginalize out the hidden POC tags g_1^L instead of the expensive sum-marginalization.

$$p(c_1^L) \cong \max_{g_1^L} p([c,g]_1^L) = \max_{g_1^L} \prod_{i=1}^{L} p([c_i, g_i] | [c,g]_{i-n+1}^{i-1}) \quad (5)$$

It is worthwhile to compare the computation of scoring for the three types of LMs, i.e. character n-grams, word n-grams, and joint n-grams. For character n-grams, scoring is straightforward, since there are no hidden variables. For word n-grams, we also need to take computations similar to Equ.(5) to score an un-segmented Chinese sentence.

Note that the scoring computation of Equ.(5) could be taken efficiently by representing n-gram LMs with WFSTs (weighted finite state transducers) [8] and performing the Viterbi decoding. For standard n-gram LMs as described in Equ.(2) (e.g. word n-grams, character n-grams), a standard algorithm for creating the WFST representation layer-by-layer is introduced in [8]. However, the revised smoothing formula for joint n-grams as in Equ.(3) are different from the standard formula as in Equ.(2). Consequently, the WFST representation for joint n-grams should also be revised correspondingly. To be precise, as mentioned in Section 2.2, only bigram smoothing formula are revised. So for higher-order joint n-gram ($n \geq 3$), the upper layers of the WFST could still be constructed according to [8].

A fragment of the revised WFST representation for an example joint n-gram LM is shown in Fig.1. For comparison, the corresponding fragment of the 'standard' (but false) WFST representation is given in Fig.2. That is, if we take it for granted to use the standard smoothing as in Equ.(2) for joint n-gram LMs, we obtain Fig.2. In both figures, only the bottom two layers of the WFSTs are shown. The main difference between the two representations is that Fig.2 has a single backoff node, while Fig.1 has four backoff nodes, corresponding to the four types of POC tags. It is notable that illegal transitions are completely removed in Fig.1 by the use of the four backoff nodes.

3. Experiments

Notations. Throughout this paper, we use "w.2g" for word bigram, "c.3g" for character trigram and "j.3g" for joint trigram. Higher-order LMs are denoted analogously.

3.1. Experimental setup for Chinese speech recognition

Evaluation experiments are carried out with a Chinese large vocabulary continuous speech recognition (LVCSR) system. Speech data for acoustic model training have a total of around 550 hours mainly obtained from LDC. The acoustic models are discriminatively-trained triphone models based on MPE (minimum phone error) criteria [9]. In the front-end, a 45-dimensional feature vector is first extracted, including 14-dimensional MFCCs with normalized log-energy and their first and second order differentials. A 3-dimensional tone feature vector is appended to the spectral features, resulting in a final feature vector of 49-dimension. Cepstral mean and variance normalization (CVN) is applied for each utterance. The test speech data is a subset (around 4 hours) from 1997 Mandarin Broadcast News Speech (HUB4-NE) released by LDC [10], with transcripts hand-segmented into words.

The evaluation metrics for Chinese LMs include perplexity and character error rate (CER) in speech recognition. To make meaningful comparison of perplexities, the length of the test speech used in the perplexity computation for word n-grams is in terms of characters instead of words. Moreover, we need some metrics to evaluate the ability of different LMs for recognizing OOV words. Note that the recognition of OOV words is not isolated and affect the recognition of the whole utterance. These metrics are better to be computed based on utterances, instead of only based on OOV words in isolation. Therefore we split the test utterances into two subsets - OOV-utterance and IV-utterance subsets. Each utterance in the OOV-utterance subset contains at least one OOV word, while the words in utterances from the IV-utterance subset are all in-vocabulary (IV) words. We then define OOV-utt-CER and IV-utt-CER as the CERs computed over OOV-utterance subset and IV-utterance subset respectively.

3.2. Speech recognition with training corpus of 1.9B characters

There are two training corpora used in our experiment. One is a smaller-scale corpus, the PKU People's Daily 1998 and 2000 with 41 million Chinese characters, which is one of the publicly available and high quality Chinese corpus with manual word segmentation. The other is a larger-scale corpus, the LDC Chinese Gigaword Fifth Edition corpus with 1.9 billion Chinese characters [11], which is not segmented into words. We first train the w.2g and j.3g LMs over the PKU corpus and apply them to segment the Gigaword corpus separately. The higher-order LMs - w.4g and j.6g LMs are separately trained on the automatically segmented Gigaword corpus based on w.2g and j.3g respectively. The c.6g is trained directly on the un-segmented Gigaword corpus. Except the joint n-gram LMs, the other LMs are all trained with the modified Kneser-Ney smoothing method by the SRILM toolkit [12].

A lexicon with 58916 words is extracted from the PKU corpus. With this lexicon, the HUB4-NE test data has an OOV-word rate (i.e. the ratio of OOV-words to the total number of words) of 2.69% and an OOV-utterance rate (i.e. the ratio of OOV-utterances to the total number of utterances) of 22.20%.

	#state	#n-gram	cut-off setting	Perplexity	Error rates (%)		
					CER	OOV-utt-CER	IV-utt-CER
Oracle	–	–	–	–	4.77	6.66	4.06
w.4g	58,916	130,118,547	0-0-1-3	28.43	20.98	23.26	20.13
c.6g	5,032	274,544,846	0-0-0-1-1-3	29.01	20.86	23.18	20.00
j.6g	15,340	299,239,752	0-0-0-1-1-3	28.71	20.84	22.83	20.10
w.4g∘c.6g	–	–	–	–	20.58	22.79	19.76
w.4g∘j.6g	–	–	–	–	20.65	22.74	19.87

Table 2: Perplexities and error rates for different LMs. #states represents the number of words, characters and joint-states respectively for w.4g, c.6g and j.6g. #n-gram represents the total number of n-grams of all orders. As an example of the terminology we use to describe cut-off settings, 0-0-1-3 means that all unigrams with 0 or fewer counts are ignored, all bigrams with 0 or fewer counts are ignored, all trigrams with 1 or fewer counts are ignored, and all fourgrams with 3 or fewer counts are ignored.

By splitting the words into characters and joint-states, the set of characters and joint-states are generated from the word-lexicon, which contain 5032 characters and 15340 joint-states respectively.

We use lattice rescoring to evaluate different LMs - w.4g, c.6g and j.6g, for speech recognition. The w.2g LM trained on the PKU corpus is used in the first-pass decoding to generate word lattices for the HUB4-NE test data, with the oracle CER 4.77%, OOV-utt-CER 6.66% and IV-utt-CER 4.06%. Next, the word lattices are transformed to character lattices, which are then rescored by different LMs. The acoustic scale factor is not tuned and fixed to be 0.08 across various LMs. The results are shown in Table 2, and the main conclusions are as follows:

1. Considering that on average there are 1.5 characters per word in Chinese [3], it is appropriate to compare a word 4-gram to a joint 6-gram. As expected, by modeling both word-level and character-level constraint, the j.6g LM outperforms both w.4g and c.6g for CER.
2. The advantage of the joint 6-gram LM is more obvious in recognizing the utterances containing OOV words. The relative reductions of OOV-utt-CERs are 1.8% and 1.5% when comparing j.6g to w.4g and c.6g respectively.
3. We examine the combination of j.6g with w.4g (denoted as w.4g∘j.6g) through weighted log-linear combination which is found to be the most useful among various combination schemes [3]. It can be seen from Table 2 that w.4g∘j.6g gives further gains over j.6g. Although w.4g∘c.6g performs close to w.4g∘j.6g in CER, w.4g∘j.6g still shows the advantage in handling OOV words, with lower OOV-utt-CERs.

4. Related work and conclusion

It is worthwhile to remark on related work on language modeling. There are recent interests in building neural network LMs (NNLMs). Feedforward NNLMs [13, 14] and recurrent NNLMs [15, 16] have been shown to yield both perplexity and word error rate improvements. The main idea is to embed words in a continuous space in which probabilities are computed via smooth functions implemented by neural networks. Its motivation is to address the problem of data sparseness and achieve better generalization for unseen n-grams. In this paper, our motivation is mainly linguistically-inspired, aiming to exploit both character-level and word-level constraints to address the OOV problem for Chinese LMs by modeling the joint-character-POC sequences. It is interesting to build NNLMs over the joint-character-POC sequences, which is a future issue.

Basically, the joint n-gram LM belongs to the feature-based LMs, like stream-based LMs [17], class-based LMs [18, 19] and factored LMs [20]. In this approach, suitable features (e.g. morphological classes, data-driven clusters, etc.) are introduced and the LMs are built over those features. Feature-based LMs have been successfully used in morphologically rich European languages [21, 22] to overcome the OOV problem. In this paper, we propose to use the POC feature which is special in Chinese and build feature-based LMs over characters. The effectiveness of modeling the joint-character-POC sequences is shown in Chinese speech recognition, especially in recognizing the utterances containing OOV words. This is the main result of this paper.

There are some related studies in Chinese LMs. First, perhaps the closest work is the character-based generative model in [23] and the joint n-gram models in [24], but both are unaware of the pitfall of naive smoothing for joint n-grams and not applied in speech recognition. A technical contribution of this paper is that the introduction of the corrected smoothing method and WFST representation for joint n-gram language modeling in speech recognition. Second, the work in [2, 3] mainly study the character n-gram LM and its combination with the word n-gram LM. But such combination is inferior to the joint n-gram LM in handling OOV words.

Finally, it can be seen from the experiments that the performance of the joint 6-gram LM may be limited by sparse estimation of the parameters. Therefore, it is interesting to find better smoothing method (e.g. neural network or log-linear approach [19]) to make full use of the modeling of the joint-character-POC sequences in the future.

5. Acknowledgements

This work is partly supported by Toshiba Corporation and Natural Science Foundation of China under grant No. 61075020. Thanks to Takashi Masuko, Pei Ding and Jie Hao for helpful discussions.

6. References

[1] Z. Dong, Q. Dong, and C. Hao, "Word segmentation needs change–from a linguist's view," in *Proc. CIPS-SIGHAN Joint Conference on Chinese Language Processing*, 2010.

[2] J. Luo, L. Lamel, and J.-L. Gauvain, "Modeling characters versus words for mandarin speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.

[3] X. Liu, J. L. Hieronymus, M. J. F. Gales, and P. C. Woodland, "Syllable language models for mandarin speech recognition: Exploiting character language models," *Journal of the Acoustical Society of America*, vol. 133, p. 519, 2013.

[4] Y.-H. Sung, M. Jansche, and P. J. Moreno, "Deploying google search by voice in cantonese," in *Proc. INTERSPEECH*, 2011.

[5] F. Peng, F. Feng, and A. McCallum, "Chinese segmentation and new word detection using conditional random fields," in *Proc. International Conference on Computational Linguistics (COLING)*, 2004.

[6] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning, "A conditional random field word segmenter for sighan bakeoff 2005," in *Proc. SIGHAN Workshop on Chinese Language Processing*, 2005.

[7] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, pp. 359–394, 1999.

[8] C. Allauzen, M. Mohri, and B. Roark, "Generalized algorithms for constructing statistical language models," in *Proc. Association for Computational Linguistics (ACL)*, 2003.

[9] D. Povey and P. C. Woodland, "Minimum phone error and ismoothing for improved discriminative training," in *Proc. ICASSP*, 2002.

[10] "1997 mandarin broadcast news speech (hub4-ne)," http://catalog.ldc.upenn.edu/LDC98S73.

[11] "Chinese gigaword fifth edition," http://catalog.ldc.upenn.edu/LDC2011T13.

[12] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Proc. INTERSPEECH*, 2002.

[13] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[14] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[15] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Proc. INTERSPEECH*, 2010.

[16] I. Sutskever, J. Martens, and G. Hinton, "Generating text with recurrent neural networks," in *Proc. ICML*, 2011.

[17] K. Kirchhoff, D. Vergyri, J. Bilmes, K. Duh, and A. Stolcke, "Morphology-based language modeling for conversational arabic speech recognition," *Computer Speech & Language*, vol. 20, no. 4, pp. 589–608, 2006.

[18] P. Brown, P. Desouza, R. Mercer, V. D. Pietra, and J. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[19] S. F. Chen, "Shrinking exponential language models," in *Proc. Human Language Technologies*, 2009.

[20] J. Bilmes and K. Kirchhoff, "Factored language models and generalized parallel backoff," in *Human Language Technology Conf. of the North American Chapter of the ACL*, 2003.

[21] G. Maltese, P. Bravetti, H. Crépy, B. Grainger, M. Herzog, and F. Palou, "Combining word-and class-based language models: a comparative study in several languages using automatic and manual word-clustering techniques," in *Proc. INTERSPEECH*, 2001.

[22] A. E.-D. Mousa, M. A. B. Shaik, R. Schlüter, and H. Ney, "Morpheme level feature-based language models for german lvcsr." in *Proc. INTERSPEECH*, 2012.

[23] K. Wang, C. Zong, and K.-Y. Su, "Which is more suitable for chinese word segmentation, the generative model or the discriminative one," in *Proc. Pacific Asia Conference on Language, Information, and Computation (PACLIC)*, 2009.

[24] X. He, Z. Ou, and J. Sun, "Joint n-gram chinese language modeling with an application to chinese word segmentation," in *Proc. IEEE International Conference on Audio, Language and Image Processing (ICALIP)*, 2012.