Structure learning in hidden conditional random fields for grapheme-to-phoneme conversion

Patrick Lehnen, Alexandre Allauzen, Thomas Lavergne, François Yvon, Stefan Hahn, Hermann Ney

Structure Learning in Hidden Conditional Random Fields for Grapheme-to-Phoneme Conversion

Patrick Lehnen[1], *Alexandre Allauzen*[2], *Thomas Lavergne*[2], *Francois Yvon*[2], *Stefan Hahn*[1], *Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, Aachen, Germany
[2]Univ. Paris-Sud, France and LIMSI/CNRS - Spoken Language Processing group
{lehnen,hahn,ney}@cs.rwth-aachen.de {allauzen,lavergne,yvon}@limsi.fr

10.21437/Interspeech.2013-544

Abstract

Accurate grapheme-to-phoneme (g2p) conversion is needed for several speech processing applications, such as automatic speech synthesis and recognition. For some languages, notably English, improvements of g2p systems are very slow, due to the intricacy of the associations between letter and sounds. In recent years, several improvements have been obtained either by using variable-length associations in generative models (*joint-n-grams*), or by recasting the problem as a conventional sequence labeling task, enabling to integrate rich dependencies in discriminative models. In this paper, we consider several ways to reconciliate these two approaches. Introducing *h*idden variable-length alignments through latent variables, our Hidden Conditional Random Field (HCRF) models are able to produce comparative performance compared to strong generative and discriminative models on the CELEX database.

Index Terms: grapheme-to-phoneme conversion, G2P, HCRF, discriminative models, hidden conditional random fields

1. Introduction

In recent years, Conditional Random Fields [1] (CRFs) have been successfully applied to several language processing applications that can be formulated as sequence labelling tasks, such as part-of-speech (POS) tagging [1], chunking [2], speech recognition [3] and language modeling [4], to name a few. Grapheme to phoneme conversion (g2p) does not so easily lend itself to CRFs, since the training data not always contain the alignment information between individual graphemic and phonemic symbols. The usual approach is to automatically compute these alignments prior to training for instance using the BIO [5] labeling scheme. However, this solution requires the alignments to be provided or computed by an external knowledge source; furthermore, the choice of a specific labeling scheme introduces an undesirable bias in the training data. As a result, g2p is artificially expressed as a sequential *l*etter classification task and fails to capture the variable-length nature of the linguistic *g*rapheme. This issue is less of a problem for generative probabilistic models, which can model hidden alignment through latent variables [6, 7], at the price of a much less restrictive set of predictive features. A natural way to get the best of both worlds is to consider Hidden CRFs [8, 9, 10, 11], which can take hidden variables into account and which was shown, in [12], to deliver state-of-the-art performances at the expense of a high computational cost. This approach remains unsatisfactory, since the g2p mappings it captures are limited, and the features it uses only consider single letters or phonemes as unit. We introduce a novel approach inspired by the phrase-based framework used in machine translation. This new HCRF model handles arbitrary mapping between graphemic and phonemic substrings since they are observed on the training data. The variable-length units is directly integrated in the model, enabling us to capture more complex dependencies. Moreover, by considering a pre-computed conversion table, the computational cost is drastically reduced.

In this paper, we compare performances achieved on the CELEX database [13] by these different kind of HCRF to a strong discriminative baseline [14] and the generative joint multigram model [6, 7]. We also describe how these approaches differ in terms of speed and the search space they explore. This paper is organised as follows: section 2 provides a short description of the HCRF approach, while section 3 proposes three implementations of this kind of models. Then features that are used by the differents models are introduced in section 4 followed by the presentation of the experimental set-up and results in section 5.

2. Hidden Conditional Random Fields

Hidden Conditional Random Fields (HCRFs) [8, 9, 10, 11] estimates the conditional probability of a phoneme sequence $y \in \mathbb{Y}_1^N$ given the observed grapheme sequence $x \in \mathbb{X}_1^M$ by considering a set of latent variable s that represents in our case the segmentation of x and y:

$$p(y|x) = \sum_{s \in \mathbb{S}} p(y, s|x) = \frac{\sum_{s \in \mathbb{S}} \exp H(y, s, x)}{\sum_{\tilde{s} \in \mathbb{S}} \sum_{\tilde{y}} \exp H(\tilde{y}, \tilde{s}, x)} \quad (1)$$

The summation over the segmentations s is restricted by a set $\mathbb{S}$ defined by the actual implementation of HCRFs. This publication will include three implementations of $\mathbb{S}$ which are described in Sec. 3. The hypotheses are ranked with the help of a general feature description $H(y, s, x) = \lambda^t h(y, s, x)$ composed of binary features $h(y, s, x) \in \{0, 1\}$ with their respective weights λ. Estimation of the parameters λ is by maximization of the conditional log-likelihood L over the training corpus $\{\overline{y}_k, x_k\}_{k=1}^K$ taking into account Elastic-Net parameter priors:

$$L(\lambda) = \sum_{k=1}^{K} \log p(\overline{y}_k|x_k) - c_1||\lambda||_1^1 - \frac{1}{2}c_2||\lambda||_2^2,$$

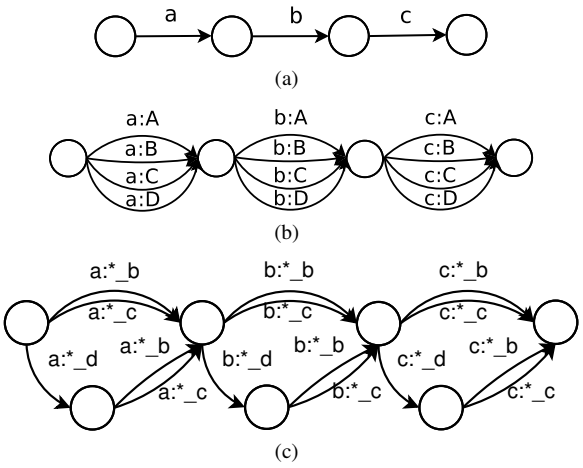

Figure 1: In System A the source symbol sequence is represented as a chain 1(a), the input chain is augmented by the target vocabulary (A, B, C, D) and weighted by prior and source-to-target features 1(b). To support segmentations s every target symbol is extended with a label representing the beginning $\_b$, the continuation $\_c$, and a doubling of a source label and beginning of a new target label $\_d$. In case the doubling was selected an alternative path is used with two arcs per source word 1(c).

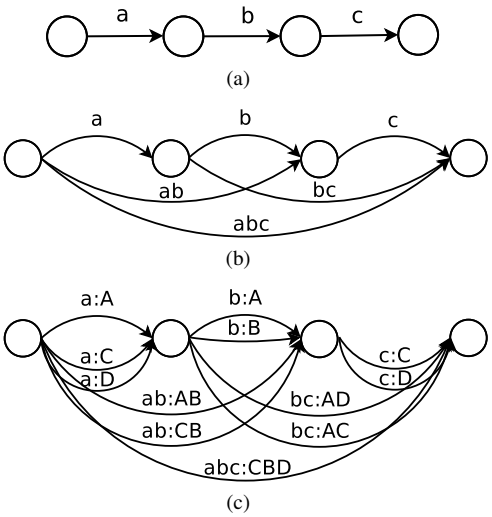

Figure 2: As in System A the utterance is represented in System B with an acceptor 2(a), which is composed with a segmentation transducer to consider all segmentations supported in the conversation table 2(b). The result is composed with the conversion table to include all conversions 2(c), and the final result is weighted with the HCRF features (Sec. 4).

where $\overline{y}_k$ denotes the reference grapheme sequence for the kth training example and c_1 and c_2 are hyperparamters that weight the regularization terms. The optimal inference rule is then to select the best sequence of phonemes $\hat{y}$ according to:

$$\hat{y} = \underset{y}{\operatorname{argmax}} \{p(y|x)\} = \underset{y}{\operatorname{argmax}} \left\{ \sum_{s \in \mathbb{S}} p(y, s|x) \right\} \quad (2)$$

The summation over all the segmentation implied by equation 2 might be computationally expensive. Therefore, the Viterbi approximation can be applied with the following inference rule:

$$\hat{y} = \underset{y}{\operatorname{argmax}} \left\{ \max_{s \in \mathbb{S}} p(y, s|x) \right\} \quad (3)$$

The objective function is optimized using resilient backpropagation, (R-PROP), a gradient descent algorithm. With the addition of a latent variable, the conditional log-likelihood is not convex like in standard CRF but expermiental result doesn't show sensitivity to the starting point.

3. Three ways to cope with hidden structure

As introduced in equation 1, the hidden variables $\mathbb{S}$ defines the search space explored by the model, i.e. the possible segmentations of the grapheme and phoneme sequences and their associations. Without any restriction, this problem is untractable. However, word internal structure and its associated pronunciation suggests that this search space can be safely restricted in some way, hence allowing exact computation of the gradient and inference. Thus, in this paper we consider three different ways to restrict the search space.

As a baseline approach (see [15, 12] for further details), an external tool provides the alignment of the training examples that are used to recast the problem as a sequence labelling task using the so-called BIO scheme [5][1]. With this assumption, the model consists of a standard linear chain CRF (LC-CRF). Its main drawbacks is that it cannot handle the case where the phoneme sequence is longer than the grapheme sequence. The second approach (System A), described in [12] and in section 3.1, proposes an extension of the BIO scheme where, $\mathbb{S}$ is by construction restricted such that any segment of graphemes can be associated to a segment of at most two phonemes. Without using an external tool to provide alignments, this restriction yields state of the art performances.

In this paper, we introduce a new model (System B) described in section 3.2 and inspired by the phrase based approach in machine translation that allows arbitrary segmentation of both the grapheme and phoneme sequences, as long as they are observed in the training set.

3.1. Description of HCRF (System A)

System A (introduced in [12]) uses phonemes annotated with special segmentation labels inspired by the *BIO* scheme [5] organized as in Hidden Markov Models (HMMs). Leaving the modeling of the phrase information to phrase-like features (Sec. 4 for more details). First, the grapheme sequence is represented as a chain of symbols in a finite state transducer [16] (Fig. 1(a)). Second each arc is duplicated for each possible phoneme symbol and weighted with the features taking only one phoneme symbol into account (Fig. 1(b)). In a third step all the phoneme symbols are extended with three labels indicating the beginning of a new phoneme $\_b$, its continuation $\_c$, and the doubling of a phoneme that implies the beginning of a phoneme $\_d$. The arcs labeled with the doubling label $\_d$ start an alternative path with two arcs per phoneme (Fig. 1(c)). The final result is composed with a n-gram acceptor weighted with the features taking only phonemes into account, and in a last step

[1] In this case, $\mathbb{S}$ is not hidden but derived from the alignments

all remaining features are applied. Known from HMMs the arcs indication continuation/beginning/doubling are weighted with a penalty $\delta_0/\delta_1/\delta_2$ (designed as HCRF features).

3.2. Description of HCRF (System B)

This system is inspired by the phrase based machine translation framework. In a preliminary step, the training corpus is aligned, e.g. with GIZA++ [17], to build the conversion table[2] that consists in a set of segments of graphemes and their associated conversion in phoneme segments. It is worth noticing that the external alignment is only used to derive this conversion table and never used by the following steps.

For inference and decoding purpose, the System B is implemented as a finite-state transducer cascade involving the following steps: the grapheme sequence is represented as an acceptor, that is composed with a segmentation transducer to consider all the segmentations; the conversion table is then applied by composition to generate the full search space that contains all the possible conversions of the input grapheme sequence; finally, the HCRF model is used to score this search space. For inference, the Viterbi hypothesis can be computed by finding the shortest path in this search space. Moreover, two additional steps can be carried on: the determinization that computes the summation of equation 2; and a composition with a n-gram model of phoneme sequence. This architecture is closely related to the proposal of [18].

4. Features

The features supporting the conditional probability $h(y, s, x) \in \{0, 1\}$ are a critical choice for CRF systems. Up to the authors knowledge, in [14] the currently best result on Celex was published. [14] use only surface features taking only the used letters on source side and the phonemes on target side into account. The authors decided to construct their systems based on the same kind of features. Basically three sets of features are used:

source-n-gram Features depending only on one target symbol y_n and a combination of source symbols $x_{s_n+\gamma_1}^{s_n+\gamma_2}$ relative to the currently aligned source word (with $\gamma_1 \leq \gamma_2$).

target-n-gram Features only describing the relation of a consecutive set of target symbols $y_{n-\delta}^{n}$ including the current target symbol y_n.

joint-n-grams Combinations of source-n-grams and target-n-grams.

System A models the search space directly via the source and target symbols. The choice of the parameters was $\gamma_1, \gamma_2 = -5, \ldots, 5$, $\gamma_1 + \gamma_2 + 1 \leq 6$, and $\delta \leq 3$. System A does not use joint-n-grams.

System B models the search space by tuples of source and target symbols, which is commonly known as phrases. Thus the smallest unit for a features is the source part of a phrase and the target part of a phrase. Using the source and target part of the phrase in the description of source-n-grams, target-n-grams and joint-n-grams as x_m and y_n, the parameters are $\gamma_1, \gamma_2 = -1, \ldots, 1$, $\delta \leq 1$ and the joint-n-grams are all combinations of the source- and target-n-grams. In average the size of a source phrase with System B is 1.84 source symbols, letting the source features span over 5.53 source symbols in average. Which is about the same size as in System A.

[2]Within the machine translation terminology, the conversion table corresponds to the set of phrase pairs or the phrase table.

5. Experimental Results

To evaluate the different approaches described in this paper, experimental results are reported using the English Celex corpus [19]. This data set is divided in three parts according to the previously published results of [20, 14] and it contains 39 995 training samples, 15 000 words for test and 5 000 word that are used as development set. The output vocabulary is made of 53 symbols (phonemes). All the results are reported in terms of both the Levenshtein based phoneme error rate (PER) and the word error rate (WER).

5.1. Build-Up of System A

For System A the regularization parameters c_1 and c_2 were optimized with respect to the error rate on the development set to $c_1 = c_2 = \frac{1}{16}$, and a save number of iterations was estimated as 75. With 75 iterations all metrics (PER, WER, number of features, conditional log-likelihood) were converged. To avoid local optima the features with respect to the currently aligned source word (y_n, x_{a_n}) were initialized with IBM-1 probabilities $\lambda = -\log(p(e|f))$. It turned out that long contexts on source and target are needed to gain good performance. The result in line 4 of Tab. 1 is poor and improves greatly with source-n-grams (n up to 6) in line 5 of Tab. 1 and are further improved with target-n-grams in line 6 and 7 of Tab. 1. In earlier experiments the need for $\delta_0/\delta_1/\delta_2$ penalties was verified, as without the penalties, e.g. the performance of line 6 of Tab. 1 is reduced to a ten times larger PER ($\approx 25\%$).

Eq. 2 suggest the use of the summation over all possible segmentations in search to be symmetric to the training conditions. Actually the full summation is computationally infeasible as it involves the determinization over a complex automaton, permitting to generate from one target symbol to two times as many target symbols as in the source sentence. We generate a k-best list with $k = 400$ and performed determinization within the k-best through summing up all equal hypotheses. As is shown in line 8 of Tab. 1 the determinization does not change the system performance significantly. As most of the time $p(y, s|x) \geq 50\%$ for the first best in the k-best list, only a small amount of the probability mass could change the recognition result.

5.2. Build-Up of System B

To build the System B, the first step aims to create the conversion table that consists in a set of segment of graphemes and their associated conversion in phoneme segments. This step is carried on by running GIZA++ with the standard setup to estimate the alignments between grapheme and phoneme sequences in both direction. Then the *grow-diag-final-and* heuristic [21] is applied to extract the conversion table. For all the results published in this article, the size of the segments on both sides (grapheme and phoneme) are limited to 3. This value was tuned experimentally on the development set.

For the decoder used during the inference and training steps, most operations directly use functionalities of the OpenFst library [22], except for the Forward-Backward algorithm and the interactions with the HCRF model, which rely on a in-house implementation[3]. The inference defined by Eq. 2 can be implemented with the determinization operation. However,

[3]The Forward-Backward algorithm can be implemented as a WFST operation using the shortest path algorithm in the log semi-ring. However, this operation being the most time consuming step in training, resorting to a tailored implementation yields a huge speed-up.

given the size of the search space, this operation can be very time consuming. For efficiency purposes, we use an approximate determinization that first extracts the k-best hypotheses, with $k = 400$. During the inference step, a n-gram model of phoneme sequence can be applied by composition after the determinization. The n-gram models are estimated with Kneser-Ney discounting [23] using all the training set.

To restrict the size of the conversion table, the segment pairs extracted from the alignment are limited to a size of 3. With this restriction the references cannot be always produced given the conversion table. A practical solution, used in many studies, is to resort to oracle references, corresponding to the best reachable solution w.r.t to a given metric, the BLEU [24] in the following experiments.

5.3. Influence of the latent variable

A popular way of modeling segmentations is to separate the estimation of the segmentation and estimation of the target sequence. As a baseline the approach based on generative joint-n-grams [20] estimated a segmentation on the training corpus respecting the target sequence reference $\overline{y}_k$. Based on the reference and the segmentation a LCCRF $p(y, s|x)$ is estimated. During search the generative joint-n-grams propose a hypothesis including a best segmentation and best target sequence. The best target sequence is omitted and only the best segmentation is used to duplicate source symbols to provide slots for the estimation of target symbols in which the LCCRF is used to find an optimal target labeling.

The resulting LCCRF (line 3 of Tab. 1) uses exactly the same features as the HCRF in line 7 of Tab. 1 except the $\delta_0/\delta_1/\delta_2$ penalties, with a significant degradation in performance with respect to PER and WER. Actually it is advantageous to model the segmentation within the CRF. Moreover, we can observe that the performances of System B are significantly worst than the others. One difference with the others is the feature set that uses segment as units instead of a single grapheme or phoneme. The other difference is the way the search space is built that makes System B drastically faster.

5.4. Comparison of the search spaces

For the sake of comparison, we provide statistics that characterize the search spaces explored by the System A and System B for two examples. For the word *aback*, System A consider a search space of 13274 nodes and 207356 arcs that corresponds to 1.74×10^{17} pathes. For the same word, System B only explores 36686 pathes with 93 nodes and 1042 arcs. For the word *bent*, System A consider a search space of 10336 nodes and 155936 arcs that corresponds to 6.22×10^{13} pathes, while the corresponding search space explored by System B contains 7914 pathes made of 92 nodes and 1016 arcs. The huge reduction of the search space explains why System B is more than a thousand times faster than System A for training and inference.

It is worth noticing that, while System A consider all the possible segmentations of the grapheme sequence, System B drastically reduces the number of segmentation by allowing a limited segment length and by considering only the segments observed in the training data. For instance, in these experiments, System B uses an inventory of 5184 grapheme segments and in average 408 different segmentations on the test set for an average word length of 8.3 graphemes. Moreover, while the System A tends to select for the test set 1-to-1 alignments (in 80% of the case) and 2-to-1 alignments (in 17% of the case), the repartition for the System B differs: 36% of 1-to-1, 29% of 2-to-2, 17% of 2-to-1, 10% of 3-to-2, 5% of 3-to-3 and 1% of 2-to-3 (the others can be neglected).

Table 1: Results on the Celex corpus. Line 1 and 2 provide baseline results where [20] is the best found generative approach, and [14] the best found discriminative approach on this task. Line 3 provides a result for a system leaving the modulation of the segmentation to a the model [20] and using the same features as in 7 except segmentation specific features (Sec. 5.3). The next two blocks describe the build-up of the System A and System B including their best result (Sec. 5.4).

			PER[%]		WER[%]	
			Dev	Eva	Dev	Eva
1		[14]				10.8
2		joint n-grams [20]		2.5		11.4
3		LCCRF	2.8	2.8	13.5	13.5
4	Sys. A	$(e_{a_j}, f_j) + (\delta_j)$	52.5	52.7	97.1	97.7
5		+ source n-grams	4.0	3.8	20.9	20.2
6		+ target-2-grams	2.6	2.5	12.6	12.3
7		+ target-3-grams	2.6	2.5	12.3	11.6
8		+ determinization	2.6	2.5	12.3	11.7
9	Sys. B			3.2		14.5
10		+ determinization		3.1		14.1
11		+ 4-gram LM		3.1		14.0

To assess, whether this restriction of the search space may explain the decrease in performance, the oracle hypothesis is estimated for the test set as explained in section 5.2. The oracle hypothesis exhibits a *PER* of 0.3% and a *WER* of 1.3%. These results show that the restriction of search space is efficient since the search space contains in average very competitive hypothesis. Unfortunately, the feature set defined at the segment level seems to be insufficient since the model is not able to select among the search space such relevant hypothesis.

6. Conclusion

In this paper we compare three ways to define Hidden Conditional Random Fields that can express a wide range of mapping between grapheme and phoneme sequences. At the expense of a degradation in performances, we introduce a model inspired by the phrase-based machine translation framework that drastically reduces the computational cost by using an efficient way to prune the search space. Experimental evidence tends to show that the poor performances of this system is due to the feature design and not to the strategy used to prune the search space. In future work, we plan to overcome this issue by a tailored feature engineering.

7. Acknowledgements

This work was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

8. References

[1] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, Williamstown, MA, USA, Jun. 2001, pp. 282–289.

[2] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 134–141.

[3] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2009.

[4] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ser. ACL '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004.

[5] L. Ramshaw and M. Marcus, "Text Chunking using Transformation-Based Learning," in *Proceedings of the 3rd Workshop on Very Large Corpora*, Cambridge, MA, USA, Jun. 1995, pp. 84–94.

[6] S. Deligne, F. Yvon, and F. Bimbot, "Variable-Length Sequence Matching for Phonetic Transcription Using Joint Multigrams," in *Proceeding of the Fourth European Conference on Speech Communication and Technology (EUROSPEECH)*, Madrid, Spain, Sep. 1995, pp. 2243–2246.

[7] M. Bisani and H. Ney, "Investigations on Joint-Multigram Models for Grapheme-to-Phoneme Conversion," in *International Conference on Spoken Language Processing*, Denver, CO, USA, Sep. 2002, pp. 105–108.

[8] A. McCallum, K. Bellare, and F. Pereira, "A conditional random field for discriminatively-trained finite-state string edit distance," in *Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. Arlington, Virginia: AUAI Press, 2005, pp. 388–395.

[9] T. Koo and M. Collins, "Hidden-variable models for discriminative reranking," in *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 507–514.

[10] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden Conditional Random Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1848–1852, 2007.

[11] X. Yu and W. Lam, "Hidden Dynamic Probabilistic Models for Labeling Sequence Data," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, Chicago, IL, USA, Jul. 2008, pp. 739–745.

[12] P. Lehnen, S. Hahn, V.-A. Guta, and H. Ney, "Hidden conditional random fields with m-to-n alignments for grapheme-to-phoneme conversion," in *Interspeech*, Portland, OR, USA, Sep. 2012.

[13] G. Burnage, "CELEX: a guide for users," University of Nijmegen, Center for Lexical Information, Nijmegen, Tech. Rep., 1990.

[14] S. Jiampojamarn, C. Cherry, and G. Kondrak, "Integrating joint n-gram features into a discriminative training framework," in *In Proceeding of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Jun. 2010, pp. 697–700.

[15] T. Lavergne, O. Cappé, and F. Yvon, "Practical Very Large Scale CRFs," in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, July 2010, pp. 504–513.

[16] M. Mohri, "Weighted automata algorithms," *Handbook of weighted automata*, pp. 213–254, 2009.

[17] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[18] S. Kumar, Y. Deng, and W. Byrne, "A weighted finite state transducer translation template model for statistical machine translation," *Natural Language Engineering*, vol. 12, no. 1, pp. 35–75, 2006.

[19] R. Baayen, R. Piepenbrock, and L. Gulikers, "Celex2," 1996.

[20] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[21] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Association for Computational Linguistics, 2003, pp. 48–54.

[22] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "Openfst: A general and efficient weighted finite-state transducer library," in *Proceedings of the 12th International Conference on Implementation and Application of Automata*, ser. CIAA'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 11–23.

[23] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 1995, pp. 181–184.

[24] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *in Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.