Integrating Joint n-gram Features into a Discriminative Training Framework

Sittichai Jiampojamarn[†] and **Colin Cherry**[‡] and **Grzegorz Kondrak**[†]

[†]Department of Computing Science
University of Alberta
Edmonton, AB, T6G 2E8, Canada
{sj,kondrak}@cs.ualberta.ca

[‡]National Research Council Canada
1200 Montreal Road
Ottawa, ON, K1A 0R6, Canada
Colin.Cherry@nrc-cnrc.gc.ca

Abstract

Phonetic string transduction problems, such as letter-to-phoneme conversion and name transliteration, have recently received much attention in the NLP community. In the past few years, two methods have come to dominate as solutions to supervised string transduction: generative joint n-gram models, and discriminative sequence models. Both approaches benefit from their ability to consider large, flexible spans of source context when making transduction decisions. However, they encode this context in different ways, providing their respective models with different information. To combine the strengths of these two systems, we include joint n-gram features inside a state-of-the-art discriminative sequence model. We evaluate our approach on several letter-to-phoneme and transliteration data sets. Our results indicate an improvement in overall performance with respect to both the joint n-gram approach and traditional feature sets for discriminative models.

1 Introduction

Phonetic string transduction transforms a source string into a target representation according to its pronunciation. Two important examples of this task are letter-to-phoneme conversion and name transliteration. In general, the problem is challenging because source orthography does not unambiguously specify the target representation. When considering letter-to-phoneme, ambiguities and exceptions in the pronunciation of orthography complicate conversion. Transliteration suffers from the same ambiguities, but the transformation is further complicated by restrictions in the target orthography that may not exist in the source.

Joint n-gram models (Bisani and Ney, 2002; Chen, 2003; Bisani and Ney, 2008) have been widely applied to string transduction problems (Li et al., 2004; Demberg et al., 2007; Jansche and Sproat, 2009). The power of the approach lies in building a language model over the operations used in the conversion from source to target. Crucially, this allows the inclusion of source context in the generative story. Smoothing techniques play an important role in joint n-gram models, greatly affecting their performance. Although joint n-gram models are capable of capturing context information in both source and target, they cannot selectively use only source or target information, nor can they consider arbitrary sequences within their context window, as they are limited by their back-off schedule.

Discriminative sequence models have also been shown to perform extremely well on string transduction problems. These begin with a Hidden Markov Model architecture, augmented with substring operations and discriminative training. The primary strength of these systems is their ability to include rich indicator features representing long sequences of source context. We will assume a specific instance of discriminative sequence modeling, DIRECTL (Jiampojamarn et al., 2009), which achieved the best results on several language pairs in the NEWS Machine Transliteration Shared Task (Li et al., 2009). The same system matches or exceeds the performance of the joint n-gram approach on letter-to-phoneme conversion (Jiampojamarn et al., 2008). Its features are optimized by an online, margin-

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 697–700,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

based learning algorithm, specifically, the Margin Infused Relaxed Algorithm, MIRA (Crammer and Singer, 2003).

In this paper, we propose an approach that combines these two different paradigms by formulating the joint n-gram model as a new set of features in the discriminative model. This leverages an advantage of discriminative training, in that it can easily and effectively incorporate arbitrary features. We evaluate our approach on several letter-to-phoneme and transliteration data sets. Our results demonstrate an improvement in overall performance with respect to both the generative joint n-gram approach and the original DIRECTL system.

2 Background

String transduction transforms an input string $\mathbf{x}$ into the desired output string $\mathbf{y}$. The input and output are different representations of the same entity; for example, the spelling and the pronunciation of a word, or the orthographic forms of a word in two different writing scripts.

One approach to string transduction is to view it as a tagging problem where the input characters are tagged with the output characters. However, since sounds are often represented by multi-character units, the relationship between the input and output characters is often complex. This prevents the straightforward application of standard tagging techniques, but can be addressed by substring decoders or semi-Markov models.

Because the relationship between $\mathbf{x}$ and $\mathbf{y}$ is hidden, alignments between the input and output characters (or substrings) are often provided in a preprocessing step. These are usually generated in an unsupervised fashion using a variant of the EM algorithm. Our system employs the many-to-many alignment described in (Jiampojamarn et al., 2007). We trained our system on these aligned examples by using the online discriminative training of (Jiampojamarn et al., 2009). At each step, the parameter update is provided by MIRA.

3 Features

Jiampojamarn et al. (2009) describe a set of indicator feature templates that include (1) context features (2) transition features and (3) linear-chain features.

context	$x_{i-c}\ y_i$ $\cdots$ $x_{i+c}\ y_i$ $x_{i-c}x_{i-c+1}\ y_i$ $\cdots$ $x_{i+c-1}x_{i+c}\ y_i$ $\cdots\cdots$ $x_{i-c}\ldots x_{i+c}\ y_i$
transition	$y_{i-1}\ y_i$
linear-chain	$x_{i-c}\ y_{i-1}\ y_i$ $\cdots$ $x_{i+c}\ y_{i-1}\ y_i$ $x_{i-c}x_{i-c+1}\ y_{i-1}\ y_i$ $\cdots$ $x_{i+c-1}x_{i+c}\ y_{i-1}\ y_i$ $\cdots\cdots$ $x_{i-c}\ldots x_{i+c}, y_{i-1}\ y_i$
joint n-gram	$x_{i+1-n}y_{i+1-n}x_iy_i$ $\cdots$ $x_{i-1}y_{i-1}x_iy_i$ $x_{i+1-n}y_{i+1-n}x_{i+2-n}y_{i+2-n}x_iy_i$ $\cdots$ $x_{i-2}y_{i-2}x_{i-1}y_{i-1}x_iy_i$ $\cdots\cdots$ $x_{i+1-n}y_{i+1-n}\ldots x_{i-1}y_{i-1}x_iy_i$

Table 1: Feature template

Table 1 summarizes these features and introduces the new set of *joint n-gram features*.

The context features represent the source side evidence that surrounds an input substring $\mathbf{x_i}$ as it generates the target output $\mathbf{y_i}$. These features include all possible n-grams that fit inside a source-side context windows of size $\mathbf{C}$, each conjoined with $\mathbf{y_i}$. The transition features enforce the cohesion of the generated output with target-side bigrams. The linear-chain features conjoin context and transition features.

The set of feature templates described above has been demonstrated to achieve excellent performance. The context features express rich information on the source side, but no feature template allows target context beyond $\mathbf{y_{i-1}}, \mathbf{y_i}$. Target and source context are considered jointly, but only in a very limited fashion, as provided by the linear chain features. Jiampojamarn et al. (2008) report that context features contribute the most to system performance. They also report that increasing the Markov order in the transition features from bigram to tri-

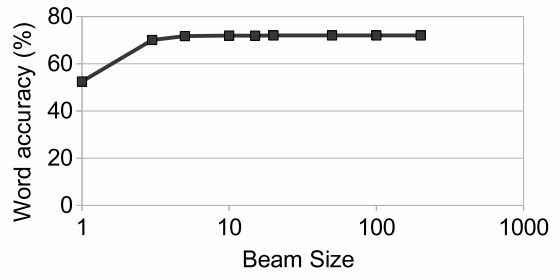

Figure 1: System accuracy as a function of the beam size

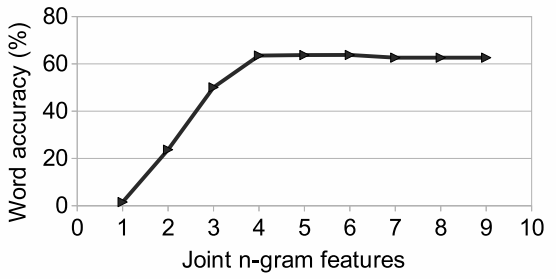

Figure 2: System accuracy as a function of n-gram size

gram results in no significant improvement. Intuitively, the joint information of both source and target sides is important in string transduction problems. By integrating the joint n-gram features into the online discriminative training framework, we enable the system to not only enjoy rich context features and long-range dependency linear-chain features, but we also take advantage of joint information between source and target substring pairs, as encoded by the joint n-gram template shown in the bottom of Table 1.

An alternative method to incorporate a joint n-gram feature would compute the generative joint n-gram scores, and supply them as a real-valued feature to the model. As all of the other features in the DIRECTL framework are indicators, the training algorithm may have trouble scaling an informative real-valued feature. Therefore, we represent these joint n-gram features as binary features that indicate whether the model has seen particular strings of joint evidence in the previous $\mathbf{n-1}$ operations when generating $\mathbf{y_i}$ from $\mathbf{x_i}$. In this case, the system learns a distinct weight for each substring of the joint n-gram.

In order to accommodate higher-order joint n-grams, we replace the exact search algorithm of Jiampojamarn et al. (2008) with a beam search. During our development experiments, we observed no significant decrease in accuracy after introducing this approximation. Figure 1 shows the system performance in terms of the word accuracy as a function of the beam size on a development set. The performance starts to converge quickly and shows no further improvement for values grater than 20. In the remaining experiments we set the beam size to 50.

We also performed development experiments with a version of the system that includes only joint n-gram indicators. Figure 2 shows the word accuracy with different values of n. The accuracy reaches its maximum for $n = 4$, and actually falls off for larger values of n. This anomaly is likely caused by the model using its expanded expressive power to memorize sequences of operations, overfitting to its training data. Such overfitting is less likely to happen in the generative joint n-gram model, which smooths high-order estimates very carefully.

4 Experiments and Results

We evaluate our new approach on two string transduction applications: (1) letter-to-phoneme conversion and (2) name transliteration. For the letter-to-phoneme conversion, we employ the English Celex, NETtalk, OALD, CMUdict, and the French Brulex data sets. In order to perform direct comparison with the joint n-gram approach, we follow exactly the same data splits as Bisani and Ney (2008). The training sizes range from 19K to 106K words. For the transliteration task, we use three data sets provided by the NEWS 2009 Machine Transliteration Shared Task (Li et al., 2009): English-Russian (EnRu), English-Chinese (EnCh), and English-Hindi (EnHi). The training sizes range from 10K to 30K words. We set $n = 6$ for the joint n-gram features; other parameters are set on the respective development sets.

Tables 2 and 3 show the performance of our new system in comparison with the joint n-gram approach and DIRECTL. The results in the rightmost column of Table 2 are taken directly from (Bisani and Ney, 2008), where they were evaluated on the same data splits. The results in the rightmost column of Table 3 are from (Jansche and Sproat, 2009), which was the best performing system based on joint

Data set	this work	DIRECTL	joint n-gram
Celex	**89.23**	88.54	88.58
CMUdict	**76.41**	75.41	75.47
OALD	**85.54**	82.43	82.51
NETtalk	**73.52**	70.18	69.00
Brulex	**95.21**	95.03	93.75

Table 2: Letter-to-phoneme conversion accuracy

Data set	this work	DIRECTL	joint n-gram
EnRu	**61.80**	61.30	59.70
EnCh	**74.17**	73.34	64.60
EnHi	**50.30**	49.80	41.50

Table 3: Name transliteration accuracy

n-grams at NEWS 2009. We report all results in terms of the word accuracy, which awards the system only for complete matches between system outputs and the references.

Our full system outperforms both DIRECTL and the joint n-gram approach in all data sets. This shows the utility of adding joint n-gram features to the DIRECTL system, and confirms an advantage of discriminative approaches: strong competitors can simply be folded into the model.

Comparing across tables, one can see that the gap between the generative joint n-gram and the DIRECTL methods is much larger for the transliteration tasks. This could be because joint n-grams are a poor fit for transliteration, or the gap could stem from differences between the joint n-gram implementations used for the two tasks. Looking at the improvements to DIRECTL from joint n-gram features, we see further evidence that joint n-grams are better suited to letter-to-phoneme than they are to transliteration: letter-to-phoneme improvements range from relative error reductions of 3.6 to 17.3, while in transliteration, the largest reduction is 3.1.

5 Conclusion

We have presented a new set of joint n-gram features for the DIRECTL discriminative sequence model. The resulting system combines two successful approaches for string transduction — DIRECTL and the joint n-gram model. Joint n-gram indicator features are efficiently trained using a large margin method. We have shown that the resulting system consistently outperforms both DIRECTL and strong joint n-gram implementations in letter-to-phoneme conversion and name transliteration, establishing a new state-of-the-art for these tasks.

Acknowledgements

This research was supported by the Alberta Ingenuity Fund and the Natural Sciences and Engineering Research Council of Canada.

References

Maximilian Bisani and Hermann Ney. 2002. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *Proc. ICSLP*, pages 105–108.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Stanley F. Chen. 2003. Conditional and joint models for grapheme-to-phoneme conversion. In *Proc. Eurospeech-2003*.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Vera Demberg, Helmut Schmid, and Gregor Möhler. 2007. Phonological constraints and morphological preprocessing for grapheme-to-phoneme conversion. In *Proc. ACL*, pages 96–103.

Martin Jansche and Richard Sproat. 2009. Named entity transcription with pair n-gram models. In *Proc. ACL-IJCNLP Named Entities Workshop*, pages 32–35.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and Hidden Markov Models to letter-to-phoneme conversion. In *Proc. HLT-NAACL*, pages 372–379.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *Proc. ACL*, pages 905–913.

Sittichai Jiampojamarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. 2009. DirecTL: a language independent approach to transliteration. In *Proc. ACL-IJCNLP Named Entities Workshop*, pages 28–31.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source channel model for machine transliteration. In *Proc. ACL*, pages 159–166.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of NEWS 2009 machine transliteration shared task. In *Proc. ACL-IJCNLP Named Entities Workshop*, pages 1–18.