INVITED PAPER

Speech-Centric Information Processing: An Optimization-Oriented Approach

The authors present a statistical framework for the end-to-end system design where the interactions between automatic speech recognition and downstream text-based processing tasks are fully incorporated and design consistency established.

By Xiaodong He, *Senior Member IEEE*, and Li Deng, *Fellow IEEE*

ABSTRACT | Automatic speech recognition (ASR) is a central and common component of voice-driven information processing systems in human language technology, including spoken language translation (SLT), spoken language understanding (SLU), voice search, spoken document retrieval, and so on. Interfacing ASR with its downstream text-based processing tasks of translation, understanding, and information retrieval (IR) creates both challenges and opportunities in optimal design of the combined, speech-enabled systems. We present an optimization-oriented statistical framework for the overall system design where the interactions between the subsystems in tandem are fully incorporated and where design consistency is established between the optimization objectives and the end-to-end system performance metrics. Techniques for optimizing such objectives in both the decoding and learning phases of the speech-centric information processing (SCIP) system design are described, in which the uncertainty in speech recognition subsystem's outputs is fully considered and marginalized. This paper provides an overview of the past and current work in this area. Future challenges and new opportunities are also discussed and analyzed.

KEYWORDS | Joint optimization; speech recognition; speech-centric information processing (SCIP); spoken language translation (SLT); spoken language understanding (SLU); voice search

I. INTRODUCTION

Automatic speech recognition (ASR) is an enabling technology for a number of important information processing applications in the realm of human language technology (e.g., [3], [4], and [35]). For example, a spoken language translation (SLT) system takes the source speech signal as input, and the output of ASR as "noisy" text is then fed into a machine translation (MT) system, producing a translated text of another target language. That is, the full SLT system can be viewed as ASR and MT subsystems in tandem (e.g., [14], [39], [62], [66], [83], and [96]). As another example, a voice search system also recognizes the input utterance as "noisy" text first, and then feeds it as a query to a subsequent information retrieval (IR) system, returning a list of documents ranked by their relevance to the query (e.g., [29], [30], and [84]). As a further example, a spoken language understanding (SLU) system again recognizes the input utterance first, and then feeds the noisy transcription to a natural language understanding (NLU) system. The NLU system will then identify the domain that the utterance represents, and/or parse the semantic meanings embedded in the utterance (e.g., [80], [85], and [87]).

In all the information processing tasks outlined above, ASR is a common component and plays a central role; hence we refer to these tasks and related applications as

Manuscript received May 2, 2012; revised September 7, 2012; accepted December 10, 2012. Date of publication March 7, 2013; date of current version April 17, 2013.

The authors are with Microsoft Research, Redmond, WA 98052 USA (e-mail: xiaohe@microsoft.com).

Digital Object Identifier: 10.1109/JPROC.2012.2236631

speech-centric information processing (SCIP). In the SCIP systems, ASR works with one (or more) downstream component(s) (i.e., the subsequent component(s) after ASR) in tandem to deliver end-to-end results. One important character in such systems is that different applications are sensitive to different errors in the ASR output. However, most of the current ASR methods embedded in SCIP systems tend to use the uniform metric of word error rate (WER) to train ASR parameters and treat all types of word errors as equally bad. Another consequence of ignoring the interactions between the subsystems is the mismatch between how the subsystems are trained and how the trained subsystems are used in the operation environment. A typical example of mismatch is the general use of large amounts of "clean" written text data to train the MT subsystem in a full SLT system while in decoding operation the MT subsystem always receives the "distorted" text input subject to ASR errors and speech disfluency. To overcome the various types of optimization inconsistency in a systematic manner and to aim for optimal design of all the subsystem components in the overall SCIP system, we need to fully incorporate the interactions between, and the uncertainty in, these subsystem components, and in particular, between ASR and MT/NLU/IR components. More specifically, we need to establish design/learning consistency between the optimization objectives and the end-to-end system performance metrics for all subsystem components.

In this paper, we will address the critical optimization inconsistency problems discussed above that are commonly present in most existing SCIP systems. This motivates the development of a unifying end-to-end optimization-oriented approach, where both the ASR and the downstream subsystems are learned via optimizing end-to-end performance metrics.

The organization of this paper is as follows. In Section II, we provide an overview of the general tandem architecture of a wide variety of SCIP systems and show how a combination of various subsystems can produce most of the common realistic systems studied and reported in the literature. In Section III, we describe and analyze the problem of optimization inconsistency inherent in most existing SCIP systems of a "divide and conquer" sort when the interactions between the subsystems are discarded. We present technical solutions to the optimization inconsistency problem in the next two sections based on a body of the published work but with generalization, unification, and new insights that cut across several types of SCIP systems. Section IV is focused on the unified objective functions for end-to-end learning of interactive SCIP subsystems' parameters. We devote Section V to the techniques for optimizing these objective functions, including a summary of experimental evidence showing the feasibility and effectiveness of these techniques. In Section VI, the experiments conducted and published in the literature that evaluate the feasibility and effectiveness of several aspects of the unified framework are reviewed and analyzed. Finally, in Section VII, we draw conclusions and discuss future directions on speech-based information processing.

II. SPEECH-CENTRIC INFORMATION PROCESSING: AN OVERVIEW

While ASR technology has important applications on its own (e.g.,[3], [4], [19], and [46]), its more significant impact lies in the combination with its downstream processing, typically referred to as human language technology, including MT, NLP, and IR [35], [68]. Interfacing ASR with one or more of the downstream information processing systems gives rise to a full SCIP system. In this section, we will first provide an architectural overview of an SCIP system. Then, we will discuss three common types of the SCIP system depending on the nature of the downstream processing.

A. Architectural Overview of SCIP Systems

In Fig. 1, we show the general tandem architecture (i.e., serial connection) that characterizes a number of SCIP systems.

Starting from the common ASR subsystem component, each path through the diagram from left to right corresponds to one specific type of the SCIP system. For instance, ASR and NLU subsystems in tandem form the SCIP system of SLU. When the output of SLU is further provided to a subsequent dialog control subsystem, a part of a spoken dialog system (open loop) is established. Similarly, when the SLT system, which consists of ASR and MT subsystems in tandem, is further connected in tandem with an NLU subsystem, we produce a cross-lingual SLU system.

Importantly, the design and learning of the diverse types of SCIP systems shown in Fig. 1 are amenable to the

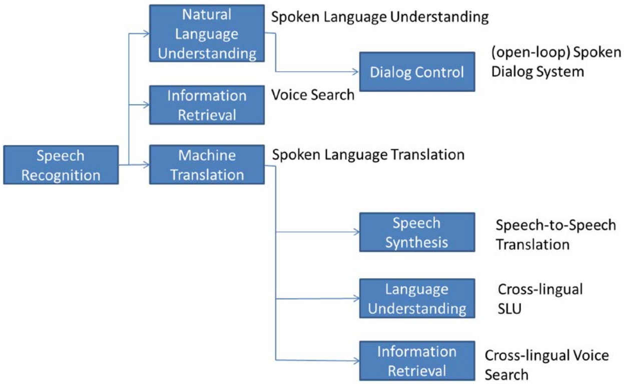

Natural Language Understanding
Spoken Language Understanding
Dialog Control
(open-loop) Spoken Dialog System
Information Retrieval
Voice Search
Speech Recognition
Machine Translation
Spoken Language Translation
Speech Synthesis
Speech-to-Speech Translation
Language Understanding
Cross-lingual SLU
Information Retrieval
Cross-lingual Voice Search

Fig. 1. *Illustration of the general tandem architecture of common SCIP systems and their relations in terms of the shared subsystem components. Design of these diverse types of systems shown here follows the shared optimization-oriented principles presented in this paper.*

common and general optimization-oriented approach to be presented and analyzed in Sections III–V. Specifically, rather than simply feeding the ASR subsystem's output directly (and unambiguously) as the input to the downstream subsystems, uncertainty of ASR, in the form of probabilistic lattices or *N*-best lists, is incorporated in the overall system's design, learning, and decoding. As we will see, taking into account the ASR uncertainty permits recovery of the errors made in ASR and this becomes essential for achieving robustness of the overall SCIP system.

Compared with the text-centric systems of NLP, IR, and MT in human language technology [35], the speech-centric systems with ASR integrated as the "front–end" present special challenges in terms of system robustness. This problem can be likened, to a certain degree, to that facing acoustic-environment robustness in ASR, which has occupied ASR research for over 25 years and is still an active research area today. Numerous techniques invented for handling environment robustness in ASR and their taxonomy have been reviewed in [22] and [53], and they are relevant to the new robustness problem in the SCIP system design arising from "noisy" text inputs (analogous to noisy acoustic inputs) due to ASR errors [102] and speech disfluency [69]. The successful techniques for noise-robust ASR aimed at achieving matched training-test acoustic environments bear resemblance to the learning strategy that exploits ASR output uncertainty. As will be explained in Sections IV and V, the use of ASR lattices or *N*-best lists in training the system parameters effectively (as well as in scoring in the decision-making phase) enhances the diversity of the "noisy" text input data to the downstream information processing subsystems. This will create a desirable learning style for the SCIP system analogous to the "multistyle" training popular in noise-robust ASR (and also analogous to the use of elastic distortion popular in training image recognition systems [78].

Owing largely to shortage of work in the literature, we intentionally exclude the prosodic modeling and speech-disfluency modeling stages in Fig. 1. They could be appropriately placed either before or after the ASR stage in Fig. 1. When appropriate modeling techniques (e.g., [88]) are used, this additional stage would also fit well in the optimization approach presented in this paper. Without including the prosodic processing/modeling stage, we simply treat the difference between what goes (as the "noisy" text input) into the downstream processing components in the SCIP system and the "clean" text input to the traditional MT, IR, or NLP systems as a combination of ASR errors and normal speech disfluency.

B. Spoken Language Translation

As shown in Fig. 1, a full SLT system can be viewed as ASR and MT subsystems in tandem (e.g., [14], [39], [62], [66], [83], and [96]). SLT is of significant relevance in our increasingly globalized world, and its research and system development started in the late 1990s (e.g., [54], [66], and [82]) after ASR had matured and become useful in practice. Following the well-established statistical framework in ASR, the same statistical approach has dominated SLT as well as MT research (e.g., [13], [14], [52], [67], [71], and [83]). In this same issue, another paper also provides a comprehensive review on latest advances in SLT [95].

The applications of SLT are diverse, ranging from machine-aided human translation [73] to professional translation services for international organizations. TC-STAR (http://www.tc-star.org) in Europe and GALE (http://www.darpa.mil/ipto/programs/gale) in the United States are the most prominent SLT research projects.

An SLT system with the ASR component to provide the input to the MT component is more difficult than text-based MT because of the compounded difficulties of ASR and MT. A particular issue in SLT is speech disfluency, making the input to the MT component of the SLT system, even with perfect ASR, deviate from lexical, syntactic, and semantic patterns of normal written texts that are typically used for training the MT system. Examples include filled pauses, paragraph and sentence delimiters, punctuation marks, and capitalized words. This deviation, together with ASR errors, produces serious "mismatch" between training and testing conditions.

One way to address this mismatch problem is to adopt the Bayesian approach where uncertainty of ASR outputs is taken into account. While the initial crude mathematical formulation of this approach appeared in the early days of SLT research [66] and later extended to joint ASR and MT decoding through an ASR lattice or confusion network [8], [96], only at the decoding stage has the ASR uncertainty been considered until rather recently when the same uncertainty was incorporated into the training process with a decision-feedback style [94]. In Section V, we will review this line of work, elaborate on how partial exploitation of ASR uncertainty at the decoding stage only can be nontrivially extended to the full exploitation (i.e., at the training state also), and provide a significantly more general and consistent framework that cuts across SLT, SLU, and other SCIP-related applications.

C. Voice Search

An ASR system followed by an IR stage produces a voice search system, as shown in Fig. 1. Voice search is the technology intended to provide users with the information they request with a spoken query [84]. The information requested often exists in a structured or unstructured large database (e.g., the Web being a huge, unstructured database). The query has to be compared with fields in the database or "documents" in the Web to obtain the relevant information. Typical voice search applications are directory assistance [1], [93] (i.e., search for the phone number and address information of a business or an individual), personal music and video management [61], infotainment in the car [77], business and product reviews [97], conference information systems [10], local search (extending directory assistance to include also maps, directions, movie schedules, local events, and travel information) [29], voice-enabled question answering, and more recently, mobile personal assistants (e.g., Siri in iPhone).

Voice search provides a convenient and direct access to a broad variety of services and information. It is particularly appealing to the users of mobile devices because of the greater efficiency to search for the desired information from the mobile devices by speech than by typing [29]. However, due to the vast amount of information available and the open nature of the spoken queries, voice search applications still suffer from both ASR and IR errors. As an example, in the voice search task of automated directory assistance, there are millions of possible business listings (over 18 million in the United States alone) as the targets for matching. Further, the users frequently do not know and say the exact business names as listed in the directory. This illustrates the special difficulty of voice search.

Typical voice search methods make use of a term frequency-inverse document frequency (TF–IDF) weighted vector space model [93], personalization features [11], analysis/parsing of input queries [27], [28], [79], and template matching [48]. In most of the above and other existing voice search work, the ASR output, subject to possible analysis, is directly fed into the IR system without considering the interactions between the two components.

A different form of voice search is called spoken document retrieval, or retrieving (and browsing) spoken content typically distributed and stored in the Web, where IR systems are deployed to access spoken "documents" produced by ASR after processing the original spoken utterances such as lecture recordings [15], [58]. The difference from voice search discussed earlier is that ASR is used here to process the stored spoken documents rather than the spoken search query. This form of voice search fits less well with the tandem architecture of SCIP shown in Fig. 1 and will not be dealt with in this paper.

D. Spoken Language Understanding

Fig. 1 also shows that when ASR and NLU subsystems are connected in tandem, the resulting pipeline gives rise to a full SLU system [18], [87], [101]. SLU has the task of mapping from an utterance to its semantic representation. In this sense, voice search just discussed can be regarded as a special form of SLU where the semantic representation is expressed in terms of the intended entry in the database or the intended document in the Web.

Traditionally, SLU tasks are divided into two broad categories. First, intent determination, also referred to as "call routing" or "How May I Help You" for historical reasons, performs the task of spoken utterance classification where the output is one of many semantic classes and there is *no sequence* information or structure at the output. Second, slot/form filling, also referred to as semantic frame-based SLU, is the task that produces the output as a *sequence* of semantic frames, with a possible hierarchical structure, from a spoken utterance [98]. Compared with intent determination, the task of slot filling generally allows a lower degree of naturalness and a smaller coverage of the language space, but it gives higher resolution or finer concepts in the output's semantic representation.

Unlike ASR (as well as MT), which accepts speech (or text) inputs in any semantic domain, current NLU technology has not been able to accomplish the task of understanding in unlimited domains [49]. Hence, the semantic space in both intent determination and slot filling of SLU is often highly constrained. This is in contrast with voice search tasks whose semantic space tends to be significantly larger.

A comprehensive coverage of slot filling, the most extensively studied SLU category, including both the traditional and more recent methods as well as technical challenges, can be found in the recent book chapter of [85]. It reviews both knowledge-based and, more importantly, data-driven statistical solutions. The latter includes generative models/methods of hidden Markov model (HMM) and composite HMM/context-free grammar (CFG) and conditional models/methods of conditional random field (CRF) and composite CFG/CRF. In the more recent work reported in [34], the results of comparative experiments are presented on three different tasks of slot filling (called concept tagging in the paper) in a set of languages with different complexity. Six methods covering both generative (finite state transducers, statistical MT, dynamic Bayesian networks) and discriminative [maximum entropy Markov models, support vector machines, conditional random fields (CRFs)] techniques are compared, and CRF turns out to be the best performing one on all tasks. Most recently, Li *et al.* have explored the multitask learning paradigm using semi-Markov CRFs on a set of slot filling tasks that overlap with each other [60].

On intent determination of SLU, a book chapter [80] also provides a comprehensive review, especially on data-driven methods. Most recently, deep learning technique has been successfully applied to intent determination, as reported by [81], [103].

E. Other SCIP Tasks

In addition to the two-component SCIP tandem systems reviewed above, Fig. 1 further shows four types of three-component SCIP tandem systems, which we briefly review here. First, when the SLU system, which consists of ASR and NLU subsystems, is further connected with a dialog control/planning component, a major part (the "open-loop" portion) of a spoken dialog system is established. Further additions of natural language generation, text-to-speech synthesis, and user modeling components will complete the full, closed-loop spoken dialog system, which has had excellent recent reviews in [90] and [91] and will not be covered in this paper. It is worth noting that the recent prevalence in mobile computing has galvanized intense and renewed interest in the work on spoken dialog systems. Some earlier, primitive form of such systems—e.g., the work on the MiPad system [21]—was limited in part by imperfection of the design and component technologies in these early days [20], but more importantly by the late arrival of mainstream mobile computing as well as the lack of a full ecosystem of Web services that Siri-style understanding and dialog systems are enjoying today.

Likewise, the two-component SLT system, composed of ASR and MT subsystems in tandem, can be extended to a three-component system by a further interface with another text-based subsystem. As shown in Fig. 1, by connecting SLT (ASR+MT) with an NLP subsystem, we produce a cross-lingual SLU system (ASR+MT+NLU) where the understanding task is now performed in a new, target language [57]. Similarly, interfacing SLT with a speech synthesis subsystem gives rise to a speech-to-speech translation system, which has applications in enabling human-to-human conversation using different languages [36], [65]. Finally, cross-lingual voice search can be accomplished when SLT is interfaced in tandem with a voice search subsystem, giving the full pipeline of ASR+MT+VoiceSearch.

We emphasize that the general design and learning principles, full exploitation of the uncertainty in the front-stage subsystems, and the optimization-oriented approach described in the remainder of this paper apply to all SCIP systems discussed in this section. However, we will mainly focus the discussions on selected, two-component systems largely due to the lack of sufficient work in the literature on other more complex SCIP systems. Specifically, we limit our discussions to the full exploitation of the uncertainty in the ASR subsystem, which is common in and essential for all types of SCIP systems.

III. OVERCOMING OPTIMIZATION INCONSISTENCY

As discussed above, ASR operates together with the downstream components to deliver the end-to-end result in any of the SCIP systems. However, optimization inconsistency that has permeated the existing design of most of the SCIP systems today is a crucial problem. In this section, we first provide an analysis on the optimization inconsistency problem from two perspectives. Then, we outline a general solution, expanded and generalized from recently published work, which overcomes the inconsistency and forms the basis of much of the remaining material in this paper.

A. The Problem of Optimization Inconsistency

SCIP is a complex information system that consists of multiple subsystems in a tandem architecture where voice-based ASR subsystem as a "front–end" is interfaced with one or more text-based subsystems including MT, NLU, and IR. Each of these subsystems has been trained using the collected supervised data with respect to the individual subsystem's own input and output signal/information. Optimization inconsistency among subsystems discussed in this section refers to the mismatch condition between the training data used to estimate the parameters of such individual subsystems and the operating environment when the decoding decision is made during the system deployment.

In conventional design, the subsystems tend to be built and trained independently, i.e., without considering the interactions between them. Sometimes, such a simplistic and easy-to-implement approach is referred to as a "divide and conquer" one and has been advocated by its proponents. Each subsystem is isolated from one another, and is assumed to take "clean" input and to produce the output results directly on its own. However, a subsystem in an actual SCIP system takes the output from the upstream subsystems as its input, and produces output that will be fed into its next downstream subsystem in tandem until the final result is delivered. Following this design philosophy, since each subsystem will necessarily produce processing errors, errors from one subsystem will propagate and impact negatively on the remaining consequent subsystem(s). That is, errors produced by the upstream subsystem at the decoding stage make the input to the downstream subsystem being polluted or "noisy." This "noisy" input mismatches the "clean" condition under which the downstream subsystem is to be trained. In this case, the output of an upstream subsystem (e.g., ASR) is just an intermediate result that will be consumed by downstream subsystems. Since this (uncertain) intermediate result is a random variable, it should be marginalized (e.g., take a summation over all possible intermediate results) in both decoding and training, a process through which the mismatch problem can be reduced.

Let us take a concrete example. In the voice search application, the ASR subsystem is most often built without considering that its recognition output will be fed into an IR subsystem, which may be able to tolerate certain types of text errors better than others. On the other hand, traditionally, an IR system is built separately, assuming the input is a relatively "clean" text, thus with little or no tolerance to the distorted text caused by ASR errors or by speech disfluency. This assumption, however, does not hold for the IR component in a full voice search system, where the IR module necessarily has to handle the output from the ASR module, which is nearly always ambiguous and error prone. The use of marginalization would enable the IR component to select possible incorrect ASR errors, as long as they can be tolerated by the IR, to strike a tradeoff with other errors that would affect IR more negatively. This kind of interactions between the subsystems is thus important to incorporate in the holistic design of the full system, which we advocate and elaborate in this paper.

In addition to the "mismatch" inconsistency just described, another important source of inconsistency in the conventional design of SCIP systems stands out between the training criteria for subsystems and the desired end-to-end evaluation metric. Historically, different downstream subsystems have had their own evaluation metrics, and the model parameters in each of such subsystems have been optimized by the objective function directly relevant to that metric. However, in the SCIP systems, different applications tend to emphasize distinct types of errors in the ASR output.

Let us again take a concrete example here. IR applications tend to focus on the match of content words, while ignoring functional words. Therefore, it is important for the ASR component to have the content words correctly recognized, while the errors in functional words can be tolerated. On the other hand, functional words bear important contextual and syntactic information, which is critical to MT. Therefore, it is crucial to recognize functional words correctly in MT applications. Unfortunately, most of the current ASR models are optimized without considering the downstream subsystems. Instead, WER is widely accepted as the *de facto* metric for ASR, treating all types of word errors equally. Since WER only measures word errors at the surface level and takes no consideration of the roles of a word in the ultimate performance measure, this often leads to suboptimal end-to-end performance. An analysis and experimental evidence for such suboptimality in the context of SLT were provided in [39], and those for the case of SLU were provided in [86].

B. End-to-End Optimization to Overcome the Inconsistency Problem

In this paper, we address the above critical optimization inconsistency problem facing the design and learning of SCIP systems. The analysis of the problem has motivated the development of a unifying end-to-end optimization framework, which fully exploits the uncertainty in each subsystem's output and the interactions between the subsystems. In this framework, the parameters of all subsystems are treated as correlating with each other and they can be trained systematically to optimize the final performance metric of the full SCIP system.

End-to-end training of SCIP systems involves optimizing difficult objective criteria [39], [41], [56], [94]. Efforts have been made and reported in the literature on the use of better optimization criteria and methods. In [42], the "margin concept" is incorporated into conventional discriminative training criteria such as minimum phone error (MPE) and maximum mutual information (MMI) for string recognition problems. In [45], a fast extended Baum–Welch (EBW) algorithm built on Kullback–Leibler (KL)-divergence-based regularization is proposed. In [50] and [51], a line search A-function (LSAF) is introduced to generalize the EBW algorithm for optimization of discriminant objective functions. In [41], a discriminative training criterion that unifies maximum mutual information (MMI), minimum classification error (MCE), and minimum phone/word error (MPE/MWE) was proposed for ASR and a growth-transformation (GT)-based optimization method for training hidden Markov model (HMM) parameters in ASR systems was presented in a systematic way. It was shown that GT-based optimization approximates the quadratic Newton update and usually gives a faster learning speed than the simple gradient-based search. More recently, in [37], this optimization method was extended to SLT based on the Bayesian framework using a similar GT- or EBW-based optimization method. In [39], experimental evidence was provided that the ASR component with the lowest WER may not necessarily lead to the best translation performance, and that global end-to-end optimization in SLT is superior to separately training ASR and MT components of an SLT system. Finally, in [94], a global end-to-end optimization for SLT was implemented using a gradient–descent technique with slow convergence. This body of work sets up the background for the more technical material in the next two sections on the establishment of optimization criteria and methods for implementing the general end-to-end learning framework. This framework and the associated optimization-oriented approach are aimed at exploiting more advanced EBW-based optimization techniques for improving global, end-to-end optimization for all types of SCIP systems. The goal here is not only faster convergence but also better performance in the overall SCIP system.

As alternatives to the EBW algorithm, other effective gradient-based methods exist [24], [63], [64]. For example, Quickprop [26] is a batch-mode optimization method. With the help of heuristics to determine the proper update step size, it approximates Newton's optimization. Rprop [75], which stands for "resilient backpropagation," is another batch-mode optimization method, which performs dynamic scaling of the update step size for each parameter based on different kinds of heuristics. In [64], a comprehensive study of gradient-based optimization methods for MCE training, including batch and semibatch probabilistic descent (PD), Quickprop, and Rprop, is given for large vocabulary speech recognition tasks. Furthermore, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method and conjugate gradient search [5], [23] are also popular gradient-based methods and are superior to other gradient–descent techniques in terms of the convergence properties. Readers are referred to [55] for further discussions.

IV. A UNIFIED UTILITY FUNCTION FOR JOINT OPTIMIZATION

While superior results were reported in earlier work on using end-to-end optimization for a variety of SCIP applications (e.g., [89] and [94]), there is a lack of a principal solution. In this section, motivated by the findings from previous work, we present a unifying solution, with solid theoretical principle, which generalizes to different types of SCIP system design and learning.

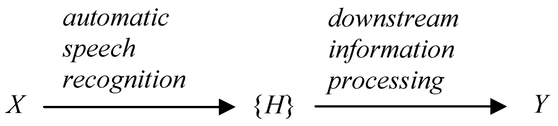


Fig. 2. ***Notations and the pipeline relations among speech signal input*** X***, the full system output*** Y***, and the intermediate ASR outputs*** $\{H\}$ ***as marginalizable hidden variables. Random variables*** $\{H\}$ ***are also the input to an MT, NLU, or IR subsystem.***

A. Notations

Without loss of generality, Fig. 2 extracts the most basic information flow in SCIP systems, where X is the observed input speech utterance, $\{H\}$ is a set of hidden random variables denoting speech recognition hypotheses that are commonly represented by a lattice or an N-best list associated with scoring information, and Y is the output from the final downstream information processing subsystem. Note that while Fig. 2 shows a tandem data flow with two subsystems, the principle and techniques presented in this and the next sections can be extended to the SCIP systems with more than two subsystems.

In the following sections, assuming there are R utterances in the training set, we denote by $\boldsymbol{X} = X_1 \ldots X_R$ the aggregate of all R training utterances. Likewise, $\boldsymbol{Y} = Y_1 \ldots Y_R$ denotes the aggregate of all R output hypotheses, one from each utterance, and $\boldsymbol{H} = H_1 \ldots H_R$ denotes the aggregate of all R recognition hypotheses, one from each utterance. In model optimization, we denote by $\boldsymbol{\Lambda}$ the set of parameters subject to optimization.

B. A Unified Utility Function

First, we define a general utility function that we would like to optimize in the joint training of SCIP systems subject to regularization. This would be the objective function for optimization if there were sufficient amounts of training data to obviate the need for adding a regularization term. Using the notations defined in Section IV-A, the utility function takes the following succinct form:

$$U(\boldsymbol{\Lambda}) = \sum_{\boldsymbol{Y}} P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}|\boldsymbol{X}) C(\boldsymbol{Y}) \tag{1}$$

where $C(\boldsymbol{Y})$ is a function tied to a classification quality measure, which scores the quality of the final output. Note that $C(\boldsymbol{Y})$ is independent of the model parameters, and it can be any arbitrary scoring function by design.

Equation (1) defines the expected quality of the end-to-end system output over the entire training corpus. In joint optimization, it is desirable to design the quality function $C(\boldsymbol{Y})$ that is close to the end-to-end evaluation metric of a particular SCIP system.

On the other hand, in order to make the computation of (1) tractable, $C(\boldsymbol{Y})$ need to be in certain decomposable form with respect to different training utterances; i.e.,

$$C(\boldsymbol{Y}) = \sum_{r=1}^{R} C(Y_r) \tag{2}$$

which states that the classification quality of the whole data set is proportional (by a constant factor of $1/R$) to the average quality of each sentence.

With the decomposition form of (2), and under the assumption that training sentences are independent of each other, the utility function can be rewritten into a tractable form

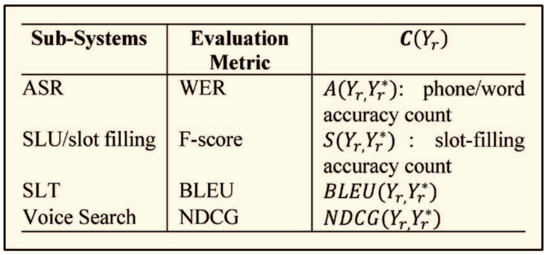
$$U(\boldsymbol{\Lambda}) = \sum_{r=1}^{R} \sum_{Y_r} P_{\boldsymbol{\Lambda}}(Y_r|X_r) C(Y_r). \tag{3}$$

A brief proof is provided here:

$$\begin{aligned} U(\boldsymbol{\Lambda}) &= \sum_{Y_1, Y_{2\ldots R}} P_{\boldsymbol{\Lambda}}(Y_1, Y_{2\ldots R}|X_1, X_{2\ldots R}) \left[C(Y_1) + \sum_{r=2}^{R} C(Y_r) \right] \\ &= \sum_{Y_1} P_{\boldsymbol{\Lambda}}(Y_1|X_1) C(Y_1) + \sum_{Y_{2\ldots R}} P_{\boldsymbol{\Lambda}}(Y_{2\ldots R}|X_{2\ldots R}) \left[\sum_{r=2}^{R} C(Y_r) \right] \\ &= \sum_{r=1}^{R} \sum_{Y_r} P_{\boldsymbol{\Lambda}}(Y_r|X_r) C(Y_r). \end{aligned}$$

Different SCIP applications have separate forms of the final output and distinct quality measures. In Table 1, we show four quality functions designed to cover the appropriate metrics for ASR and three speech-centric applications: SLU, SLT, and voice search, where Y_r^* denotes the target reference of the rth input sentence. As an example, in SLT, the final output Y is a sentence in the target language, and the quality of Y is commonly measured by the bilingual evaluation understudy (BLEU) score [70] given the reference translation Y^*. On the other hand, in voice search, each ASR hypothesis H is fed into the IR system as a query, and the final output Y is a list of ranked

Table 1 Four Quality Functions That Correspond to Appropriate Evaluation Metrics for ASR, SLU, SLT, and Voice Search



Sub-Systems	Evaluation Metric	$C(Y_r)$
ASR	WER	$A(Y_r, Y_r^*)$: phone/word accuracy count
SLU/slot filling	F-score	$S(Y_r, Y_r^*)$: slot-filling accuracy count
SLT	BLEU	$BLEU(Y_r, Y_r^*)$
Voice Search	NDCG	$NDCG(Y_r, Y_r^*)$

documents that are retrieved from a document set. The quality of Y is usually measured by comparing it against the gold set of documents Y^*, sorted by the relevance to the original spoken query as judged by humans. In IR, one widely adopted metric is the normalized discounted cumulative gain (NDCG) score [47].

By taking different forms of the classification quality measure function $C(Y_r)$, the unified utility function encompasses a range of SCIP systems. Note that the corpus level quality measure is $C(\boldsymbol{Y}) = \sum_{r=1}^{R} C(Y_r)$.

For some applications, the evaluation metric is scored over the whole data set and thus cannot be decomposed directly. Examples are the F-measure for the slot filling task of SLU [98] and then the BLEU score for SLT. In this case, we need to design a decomposable quality function that approximates the true metric. For example, the sentence-level BLEU is used to approximate the corpus-level BLEU. In practice, we found the sentence-level BLEU correlates well with the corpus-level BLEU [38].

If the parameter set of the downstream subsystem is not accessible, we can treat that subsystem as a fixed black box, and train other subsystems jointly with the end-to-end system performance as the objective. As an example, if the commercial online web search service is used as the back–end of a voice search system, where the commercial search service is provided as is, we want to optimize the ASR system so that the end-to-end voice search performance is optimized. In this case, we can design the utility function as follows:

$$U(\boldsymbol{\Lambda}) = \sum_{\boldsymbol{H}} P_{\boldsymbol{\Lambda}}(\boldsymbol{H}|\boldsymbol{X}) \left[\sum_{r=1}^{R} NDCG\left(Y_r, Y_r^*\right) \right] \tag{4}$$

where $Y_r = IR(H_r)$ is the ranked document list retrieved through feeding the speech recognition hypothesis H_r to the back–end IR system. This utility function effectively scores the expected quality of the ASR output, which is measured by the IR performance resulting from using the recognition hypothesis as query. This gives a special case of the general utility function of (1).

The utility function of (1) provides a principled and practical way of constructing the optimization objective, and has four important merits. First, the evaluation metrics of most applications are not smooth. In earlier work, the metric had to be modified to make it differentiable so as to facilitate model training. In contrast, the utility function of (1) is independent of the model parameters and can take a more flexible form. Second, the conventional discriminative training methods require a target reference, and the model parameters are adjusted such that the system will produce outputs that approach that reference. However, in complex tasks such as MT, specifying a true reference is difficult and often the true reference may not be reachable [100]. For the utility function of (1), there is no need to explicitly specify a discriminative reference or pseudoreference target. Third, the utility function of (1) is directly linked to the end-to-end evaluation metric, minimizing the discrepancy between the training criterion and the evaluation metric. Fourth, the utility function of (1) is in a form suitable for the use in extended Baum–Welch (EBW) optimization algorithm, which is efficient and scalable to handle large data sets. Moreover, EBW reestimation formulas can often provide useful insight on how the parameters are influenced by each other during the optimization process. They also offer intuitive interpretations for the model updating process. This is particularly important for the analysis of the complex interactions of subsystems and their impact on model estimation for the SCIP systems. Concrete examples will be given on the EBW formulas and their interpretations in the next section.

C. Modeling $P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}|\boldsymbol{X})$ in the Utility Function

To complete the specification of the utility function of (1), here we model end-to-end SCIP systems by a general log-linear model, where the interactions between the subsystems are jointly modeled.

Given the speech signal X, the final output of an SCIP system Y is decoded by

$$\hat{Y} = \arg\max_{Y} P_{\boldsymbol{\Lambda}}(Y|X). \tag{5}$$

When we view the recognition hypothesis H as a hidden structure between X and Y, then according to the law of total probability, we have

$$P_{\boldsymbol{\Lambda}}(Y|X) = \sum_{H} P_{\boldsymbol{\Lambda}}(Y,H|X) \approx \max_{H} P_{\boldsymbol{\Lambda}}(Y,H|X) \tag{6}$$

and when the downstream subsystem is modeled by a log-linear model, we can also represent the posterior probability of the (Y,H) pair given X through a log-linear model as follows:

$$P_{\boldsymbol{\Lambda}}(Y,H|X) = \frac{1}{Z}\exp\left\{ \sum_{i} w_i \log \varphi_i(Y,H,X) \right\} \tag{7}$$

where $Z = \sum_{Y,H} \exp\{\sum_i w_i \log \varphi_i(Y,H,X)\}$ is a normalization denominator to ensure $P_{\Lambda}(Y,H|X)$ sum to one over the space of the (Y,H) pair, $\boldsymbol{w} = \{w_i\}$ are feature weights, and $\{\varphi_i(Y,H,X)\}$ are the feature functions, also called component models, empirically constructed to capture the dependency among Y, H, and X. For simplification, we denote by $\boldsymbol{\Gamma}$ the set of parameters of all feature functions subject to optimization, and the complete parameter set $\boldsymbol{\Lambda} = \{\boldsymbol{w}, \boldsymbol{\Gamma}\}$. In the following sections, we may use $\boldsymbol{\Lambda}$, $\boldsymbol{w}$, and $\boldsymbol{\Gamma}$ to represent parameters as appropriate to emphasize the parameter set that is subject to optimization within the current context.

Conventionally, it is assumed that H depends only on X through the ASR subsystem and Y depends only on H through the downstream subsystem. Then, the feature set of the overall SCIP system is a mere collection of the ASR model (e.g., HMM and language model) and component models in the downstream subsystem. Moreover, once being integrated through (7), models from different subsystems will compete and/or support each other to estimate the integrated score of (6) for each hypothesis. This integrated score ensures that the decoding process is able to deliver a global optimal output incorporating the interactions between the subsystems. Using SLT as an example, we now elaborate on this joint modeling framework below.

The ASR component in an SLT system is commonly modeled through a noisy-channel model; i.e., the posterior of the recognition hypothesis given the speech signal is

$$P(H|X) \propto P(X|H)P(H) \tag{8}$$

where $P(X|H)$ is usually represented by an HMM-based acoustic model, and $P(H)$ by an n-gram language model (LM) for the source language. However, the actual decoding process in ASR practice is

$$\arg\max_{H}[\log P(X|H) + w_{\mathrm{LM}} \log P(H) + w_{\mathrm{WC}}|H|] \tag{9}$$

where w_{LM} and w_{WC} are the LM scale and the word count scale. Therefore, we can construct ASR-relevant feature functions as follows:

$$\varphi_{\mathrm{AM}} = P(X|H) \quad \varphi_{\mathrm{LM}} = P(H) \quad \varphi_{\mathrm{WC}} = e^{|H|}.$$

Modern MT is commonly represented by a log-linear model [67]. For example, the widely adopted phrase-based MT has features including an n-gram target language model, a reordering model, source-to-target and target-to-source phrase translation models, source-to-target and target-to-source lexicon translation models, target word counts, and phrase counts.

In SLT, it is usually assumed that the MT process depends on the input speech only through the recognition hypothesis. Hence, the features for both ASR and MT components are simply aggregated to form the feature set in (7) [14], [39]. Nevertheless, it is worth noting that (7) enables the possibility of developing and integrating more informative features $\varphi_i(Y, H, X)$ capable of capturing the dependency between speech input and translation output directly. A potential direction is to use prosodic features in this regard. The prosody of speech (in the source language side of SLT) bears important information that is potentially helpful for translation. For instance, prosody can help to more accurately translate emotion expressions of the user. Unfortunately, in most of the current SLT systems, the prosodic information after ASR is lost. Given the joint modeling framework discussed here, it is possible to design features $\varphi_i(Y, H, X)$ that embed the prosodic influence and enable appropriate dependency between speech input and translation output.

Equation (7) is defined on a single sentence; however, it is straightforward to extend it to the full training corpus, yielding

$$P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}, \boldsymbol{H}|\boldsymbol{X}) = \frac{1}{Z}\exp\left\{\sum_i w_i \log \varphi_i(\boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{X})\right\} \tag{10}$$

where the features of the full corpus are constructed by

$$\varphi_i(\boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{X}) = \prod_{r=1}^{R} \varphi_i(Y, H, X). \tag{11}$$

Accordingly, at the full-corpus level, we have

$$P_{\boldsymbol{\Lambda}}(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{\boldsymbol{H}} P_{\Lambda}(\boldsymbol{Y}, \boldsymbol{H}|\boldsymbol{X}). \tag{12}$$

Equations (10)–(12) describe the actual models needed in computing the utility function of (1). In Section V, we will discuss the techniques for jointly estimating the ASR and the downstream subsystems' parameters $\boldsymbol{\Gamma}$ in these models, as well as the feature weights $\boldsymbol{w}$, so as to optimize (1).

V. TECHNIQUES FOR JOINT OPTIMIZATION

The complete parameters that are subject to optimization in an SCIP system consist of two sets: $\{\boldsymbol{w}, \boldsymbol{\Gamma}\}$, where $\boldsymbol{w} = \{w_i\}$ are the feature weights in the log linear model of (10), and $\boldsymbol{\Gamma}$ are the parameters that characterize the feature functions of $\varphi_i(\boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{X})$ in (10). Note that exactly what $\boldsymbol{\Gamma}$ entails is dependent upon how the subsystems (e.g., ASR, SLT, NLU, and IR) are parametrically modeled. In this section, we first describe the optimization techniques for the parameter sets $\boldsymbol{w} = \{\boldsymbol{w}_i\}$ and $\boldsymbol{\Gamma}$, respectively. Then, we describe the complete training procedure that iteratively trains sets $\boldsymbol{w} = \{w_i\}$ and $\boldsymbol{\Gamma}$. Joint optimization in the section title here refers to the joint parameters in the feature function of $\varphi_i(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})$, which contains the free parameters in both ASR and its downstream processing subsystems, the main topic in Section V-B.

A. Learning Feature Weights in the Log-Linear Model

The size of free parameters of the log-linear model, i.e., the set of feature weights denoted by $\boldsymbol{w} = \{w_i\}$ in (10), is usually small. These parameters can be trained by directly maximizing the utility function or the evaluation metric associated with the final output of the SCIP system on a development set; i.e.,

$$\widehat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}} \mathrm{Eval}\left(\widehat{\boldsymbol{Y}}(\boldsymbol{w},\boldsymbol{X}), \boldsymbol{Y}^*\right) \tag{13}$$

where $\boldsymbol{Y}^*$ is the reference, and $\widehat{\boldsymbol{Y}}(\boldsymbol{w},\boldsymbol{X})$ is the final system output obtained through the decoding process according to (5) given input $\boldsymbol{X}$ and (initialized) feature weights $\boldsymbol{w}$. Eval() is the evaluation metric, shown in the second column of Table 1 for various SCIP systems, which scores the quality of $\widehat{\boldsymbol{Y}}$. When the number of weights is relatively small, the weights $\boldsymbol{w} = \{w_i\}$ are usually tuned by methods such as minimum error rate training (MERT) [67] and Powell's search or hill climbing [12]. When there are a large number of such feature weights, since the evaluation-metric-derived training objective is usually not convex, numerical optimization algorithms such as perceptron and margin infused relaxed algorithm (MIRA) are often used as reported in the literature [99], [100].

B. Learning Joint Parameters Inside the Feature Functions

Compared with feature weights $\boldsymbol{w}$, the number of parameters $\boldsymbol{\Gamma}$ in the feature functions or component models $\varphi_i(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})$ in (10) is typically much larger. For example, there are hundreds of thousands of multivariate Gaussian models in a modern acoustic model, and millions of n-grams in a language model. Therefore, learning parameters of these models presents a significant challenge. In brief, there are two major problems when designing the learning method. First, given the large number of free parameters, proper regularization is important to achieve robust parameter estimation. Second, in order to learn the many free parameters, large-scale training materials are necessary; hence, efficiency and scalability in the optimization technique are critical.

Below we will present parameter regularization, followed by the application of an efficient and scalable method based on EBW algorithm for optimizing the parameters in feature functions.

1) Regularization and the Training Criterion: As a powerful technique in machine learning, regularization is applied to control the complexity of the model, where the most common regularization methods are based on the norm of the parameters [9]. However, for the model of (7), since most of the component models are probabilistic, KL-divergence-based regularization also fits the need well. KL regularization has been studied in the machine learning community. The study of [2] uses KL regularization for sparse coding, and shows that KL regularization retained the desirable pseudo-*sparsity* characteristics of L1 regularization while being differentiable. In training SCIP-related systems, KL regularization effectively prevents the new parameters from being too far away from a constant prior model, which was found effective experimentally [38].

We encounter both continuous-density Gaussian models (e.g., in the acoustic model of ASR) and discrete distribution models in common speech-centric information systems (e.g., transition probabilities of HMM and language models for ASR and many types of distributions in MT, IR, and NLU models). For the Gaussian distributions in ASR, the KL regularization is defined as

$$\mathrm{KL}(\psi^0\|\psi) = \sum_i \mathrm{KL}\left(N_i^0\|N_i\right) \tag{14}$$

where we denote by ψ the set of Gaussians and N_i the ith Gaussian distribution, e.g.,

$$p(x;\mu,\Sigma) \propto |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right). \tag{15}$$

The KL divergence between two Gaussians is

$$\mathrm{KL}(N^0\|N) = \frac{1}{2}\Bigg(\mathrm{tr}(\Sigma^{-1}\Sigma_0) + (\mu-\mu_0)^T\Sigma^{-1}(\mu-\mu_0) - \ln\left(\frac{\det\Sigma_0}{\det\Sigma}\right) - k\Bigg). \tag{16}$$

On the other hand, we denote by $\boldsymbol{\theta}$ the set of all parameters in discrete distributions. To simplify the notation, $\boldsymbol{\theta}$ is formed as a matrix, where its elements $\{\theta_{ij}\}$ are probabilities subject to $\Sigma_j\theta_{ij} = 1$, e.g., each row is a probability distribution.

The KL regularization of discrete distributions is defined as the sum of KL divergence over the entire discrete parameter space

$$\mathrm{KL}(\boldsymbol{\theta}^0\|\boldsymbol{\theta}) = \sum_i\sum_j \theta_{ij}^0 \log\frac{\theta_{ij}^0}{\theta_{ij}}. \tag{17}$$

Then, the overall KL regularization becomes

$$\mathrm{KL}(\boldsymbol{\Gamma}^0\|\boldsymbol{\Gamma}) = \mathrm{KL}(\boldsymbol{\psi}^0\|\boldsymbol{\psi}) + \mathrm{KL}(\boldsymbol{\theta}^0\|\boldsymbol{\theta}). \tag{18}$$

Given the regularization, the objective function to be maximized in training is

$$O(\boldsymbol{\Gamma}) = \log U(\boldsymbol{\Gamma}) - \tau \cdot \mathrm{KL}(\boldsymbol{\Gamma}^0\|\boldsymbol{\Gamma}) \tag{19}$$

where the prior model $\boldsymbol{\Gamma}^0$ can take the maximum-likelihood trained model without joint discriminative training. τ is a scaling factor controlling the regularization term, e.g., $\tau = 0$ results in no regularization. In practice, different values of τ could be assigned to $\mathrm{KL}(\boldsymbol{\psi}^0\|\boldsymbol{\psi})$ and $\mathrm{KL}(\boldsymbol{\theta}^0\|\boldsymbol{\theta})$ to accommodate the difference between dynamic ranges of KL distances of continuous distribution and discrete distribution, respectively.

We now describe how the objective function of (19) can be optimized.

2) Basics of the EBW Algorithm: Here we briefly review the EBW algorithm and demonstrate how it can be applied to optimizing some specific forms of the objective function.

Baum–Eagon inequality [6], [7] gave the parameter estimation formula to iteratively maximize positive-coefficient polynomials of random variables that are subject to sum-to-one constants. Gopalakrishnan *et al.* [32] extended the algorithm to handle rational functions, i.e., a ratio of two polynomials, which is commonly encountered in discriminative training.

Consider a set of random variables $\mathbf{p} = \{p_{ij}\}$ that are subject to the constraint of $\Sigma_j p_{ij} = 1$. Assume that $g(\mathbf{p})$ and $h(\mathbf{p})$ are two positive polynomial functions of $\mathbf{p}$. Then, a growth transformation (GT) of $\mathbf{p}$ for the rational function $r(\mathbf{p}) = g(\mathbf{p})/h(\mathbf{p})$ can be obtained through the following two steps, which will iteratively optimize the value of $r(\mathbf{p})$.

1) Construct the auxiliary function

$$f(\mathbf{p}) = g(\mathbf{p}) - r(\mathbf{p}')h(\mathbf{p}) \tag{20}$$

where $\mathbf{p}'$ are the values from the previous iteration. Increasing f guarantees an increase of r, i.e., $h(\mathbf{p}) > 0$ and $r(\mathbf{p}) - r(\mathbf{p}') = (1/h(\mathbf{p})) \times (f(\mathbf{p}) - f(\mathbf{p}'))$.

2) Derive GT formula for $f(\mathbf{p})$

$$p_{ij} = \frac{p'_{ij}\left.\dfrac{\partial f(\mathbf{p})}{\partial p_{ij}}\right|_{\mathbf{p}=\mathbf{p}'} + D \cdot p'_{ij}}{\displaystyle\sum_j p'_{ij}\left.\dfrac{\partial f(\mathbf{p})}{\partial p_{ij}}\right|_{\mathbf{p}=\mathbf{p}'} + D} \tag{21}$$

where D is a smoothing factor.

The EBW algorithm was originally proposed for discriminative learning of discrete distributions. Later, Axelrod *et al.* [16], Gunawardana and Byrne [33], and Normandin [76] extended it to discriminatively train continuous density distributions such as Gaussian models, leading to substantial success in ASR [41], [72].

3) Learning Discrete Distributions: The EBW algorithm can be applied to learn discrete feature parameters by optimizing the objective function (19). Since maximizing $O(\boldsymbol{\theta})$ is equivalent to maximizing $e^{O(\boldsymbol{\theta})}$, we transform the original objective function $O(\boldsymbol{\theta})$ into the following objective function:

$$R(\boldsymbol{\theta}) = U(\boldsymbol{\theta})e^{-\tau\cdot\mathrm{KL}(\boldsymbol{\theta}^0\|\boldsymbol{\theta})}. \tag{22}$$

In order to optimize $\boldsymbol{\theta}$, i.e., the set of discrete parameters, we substitute (1), (10), (12), and (18) into (22), drop terms that are irrelevant to optimizing $\boldsymbol{\theta}$, and obtain $R(\boldsymbol{\theta})$ in a rational function form (see the derivation steps in Appendix I)

$$R(\boldsymbol{\theta}) = \frac{G(\boldsymbol{\theta}) \cdot J(\boldsymbol{\theta})}{H(\boldsymbol{\theta})} \tag{23}$$

where

$$\begin{aligned} G(\boldsymbol{\theta}) &= \sum_Y \sum_H \sum_i \varphi_i^{w_i}(\boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{X})C(\boldsymbol{Y}) \\ J(\boldsymbol{\theta}) &= \prod_i \prod_j \theta_{ij}^{\tau\theta_{ij}^0} \\ H(\boldsymbol{\theta}) &= \sum_Y \sum_H \prod_i \varphi_i^{w_i}(\boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{X}) \end{aligned}$$

are all positive polynomials of $\boldsymbol{\theta}$. Therefore, we can follow the two steps of EBW to derive the GT formulas for $\boldsymbol{\theta}$.

We now use SLT as a concrete example to discuss the EBW reestimation formula for the phrase translation model of MT in the remaining part of this section. In phrase-based translation, the input sentence is segmented into K

phrases, and the source-to-target forward phrase (FP) translation feature is scored as

$$\varphi_{\mathrm{FP}}(H,Y,X)=\prod_k p(\bar{y}_k|\bar{h}_k) \tag{24}$$

where $\bar{h}_k$ and $\bar{y}_k$ are the kth phrase in the recognition hypothesis H and translation hypothesis Y, respectively.

As shown in Table 1, $C(\boldsymbol{Y})$ takes the form of $\Sigma_{r=1}^{R}\mathrm{BLEU}(Y_r,Y_r^*)$ for SLT. Using the EBW derivation steps provided in Appendix II, we obtain the reestimation formula for updating the following parameters (probability of translating the source phrase i to the target phrase j):

$$p_{ij}=\frac{\sum_r\sum_{H_r}\sum_{Y_r}\gamma_{\mathrm{FP}}(H_r,Y_r,r,i,j)+U(\boldsymbol{\theta}')\tau_{\mathrm{FP}}p_{ij}^0+D_ip_{ij}'}{\sum_j\sum_r\sum_{H_r}\sum_{Y_r}\gamma_{\mathrm{FP}}(H_r,Y_r,r,i,j)+U(\boldsymbol{\theta}')\tau_{\mathrm{FP}}+D_i} \tag{25}$$

where $\boldsymbol{\theta}'$ is the model obtained from the immediately previous iteration $\tau_{\mathrm{FP}}=\tau/w_{\mathrm{FP}}$, and

$$\gamma_{\mathrm{FP}}(H_r,Y_r,r,i,j)=P_{\boldsymbol{\theta}'}(Y_r,H_r|X_r)\cdot\left[\mathrm{BLEU}\left(Y_r,Y_r^*\right)-U_r(\boldsymbol{\theta}')\right]\cdot\sum_k\mathbf{1}(\bar{h}_{r,k}=i,\bar{y}_{r,k}=j) \tag{26}$$

in which the utility function $U_r(\boldsymbol{\theta}')$ is the expected BLEU score for sentence r using models from the previous iteration; i.e.,

$$U_r(\boldsymbol{\theta}')=\sum_{Y_r}P_{\boldsymbol{\theta}'}(Y_r|X_r)\mathrm{BLEU}\left(Y_r,Y_r^*\right). \tag{27}$$

The smoothing factor set of D_i according to the Baum–Eagon inequality is usually far too large for practical use [32]. One general guide for empirically setting the smoothing factor D_i is to make all updated probabilities positive. Following [38], we compute

$$D_{i,\mathrm{den}}=\sum_j\sum_r\sum_{H_r}\sum_{Y_r}\max(0,-\gamma_{\mathrm{FP}}(H_r,Y_r,r,i,j)) \tag{28}$$

which ensures that the denominator of (25) is positive. We also compute

$$D_{i,\mathrm{nor}}=\max_j\left\{\frac{-\sum_r\sum_{H_r}\sum_{Y_r}\gamma_{\mathrm{FP}}(H_r,Y_r,r,i,j)}{p_{ij}'}\right\} \tag{29}$$

that guarantees that the numerator is positive also. Then, D_i is set to be the maximum of these two values

$$D_i=\max\{D_{i,\mathrm{nor}},D_{i,\mathrm{den}}\}. \tag{30}$$

To gain insight into the desirable properties of the EBW reestimation formula of (25), let us first compare the phrase model's training formula of SLT and that of regular text-based MT. That is, if the recognition hypothesis is replaced by the true speech transcription H_r^*, SLT is reduced to MT and so should the related EBW reestimation formulas. This can be verified by analyzing the model updating formula in (25) and (26). To this end, we first eliminate summation over H_r in (25). Then, since H_r^* is a deterministic (true) transcription of X_r, we have $P_{\theta'}(H_r^*|X_r)\equiv 1$. This leads to

$$P_{\boldsymbol{\theta}'}\left(Y_r,H_r^*|X_r\right)=P_{\boldsymbol{\theta}'}\left(Y_r|H_r^*\right)P_{\boldsymbol{\theta}'}\left(H_r^*|X_r\right)=P_{\boldsymbol{\theta}'}\left(Y_r|H_r^*\right).$$

Thus, the EBW reestimation formula of (25) for SLT is reduced to

$$p_{ij}=\frac{\sum_r\sum_{Y_r}\gamma_{\mathrm{FP}}\left(H_r^*,Y_r,r,i,j\right)+U(\boldsymbol{\theta}')\tau_{\mathrm{FP}}p_{ij}^0+D_ip_{ij}'}{\sum_j\sum_r\sum_{Y_r}\gamma_{\mathrm{FP}}\left(H_r^*,Y_r,r,i,j\right)+U(\boldsymbol{\theta}')\tau_{\mathrm{FP}}+D_i} \tag{31}$$

which is exactly the same as the EBW reestimation formula developed in [38] for text-based MT.

Further insight can be gained by comparing (31) for MT to (25) for SLT to appreciate how the ASR's behavior is automatically taken into account when training the phrase translation model for SLT. It is clear from (25) that the estimator in such jointly trained SLT considers possible phrases in all ASR hypotheses as potential source phrases. These include the phrases in incorrect ASR outputs, which nevertheless may result in good translation as driven by the right utility function. This desirable property becomes even clearer in (26), which computes a modified fractional count for the phrase pair. According to (26), the fractional count will be positive if the resulting translation is good, as measured by its BLEU score being better than average. This is consistent with the intuition about a good estimator. The actual value of the fractional count depends also on how likely the translation is (conditioned on the recognition hypothesis), which is measured by $P_{\theta'}(Y_r|H_r)$, and how likely the recognition hypothesis is, which is measured by $P_{\theta'}(H_r|X_r)$. All these intuitive dependencies are reflected in (26) through the factor of $P_{\theta'}(Y_r,H_r|X_r)=P_{\theta'}(H_r|X_r)P_{\theta'}(Y_r|H_r)$. Therefore, the EBW estimate automatically implements this desirable and intuitive notion:

as long as a particular translation is reasonably accurate (better than average and not necessarily the top one), all phrase pairs that contribute to this translation, as denoted by $\Sigma_k \mathbf{1}(\bar{h}_{r,k} = i, \bar{y}_{r,k} = j)$ in (26), will receive positive fractional counts according to (26). The size of the counts is the product of the likelihood of the translation given the recognition hypothesis and the likelihood of the hypothesis given the speech utterance (not necessarily the most accurate ASR hypothesis), according also to (26). Then, such positive fractional counts help boost the translation probability of that phrase pair according to (25).

Similar EBW reestimates are derived for other discrete models, such as the *n*-gram language model, and the discrete parameters in the feature functions used in the SLU, voice search, and other SCIP systems. These reestimation formulas also offer intuitive interpretations in their respective application domains, just like the interpretations provided to the phrase translation probability of SLT as detailed above.

4) Learning Gaussian Distributions in ASR: Using SLT as an example again and following similar derivation steps to those presented in [41] and [37], we establish a set of EBW reestimation formulas for the Gaussian parameters in the Gaussian-mixture HMM-based speech recognition subsystem within any SCIP system. Taking the mean vector of the *i*th Gaussian distribution as an example, we write down the reestimation formula as (32), shown at the bottom of the page, where

$$\gamma_G(H_r, Y_r, r, i, t) = P_{\psi'}(Y_r, H_r|X_r) \cdot \left[\mathrm{BLEU}\left(Y_r, Y_r^*\right) - U_r(\psi')\right] \cdot \gamma_{i,H_r}(t) \qquad (33)$$

in which $U_r(\psi')$ is computed similarly to (27), and

$$\gamma_{i,H_r}(t) = P_{\psi'}(q_{r,t} = i|X_r, H_r) = \sum_{q:q_t=i} P_{\psi'}(q|X_r, H_r) \qquad (34)$$

is the occupation probability of HMM state *i* at time *t* of the *r*th sentence.

By analyzing the model update formulas of (32) and (33), it is clear that the Gaussian means in the ASR model are trained to avoid producing recognition hypotheses that may lead to poor translation for SLT (or poor understanding for SLU, or poor IR in voice search in which cases, BLEU would be replaced by F-measure or NDCG, respectively). Here is why: According to (33), the modified fractional count γ_G will take a large negative value if $P_{\psi'}(Y_r, H_r|X_r)$ is large and the resulting translation has a low or at least below-average BLEU score. On the other hand, the model parameters will not be penalized much for producing recognition errors as long as the resulting translation quality is not affected much (or staying about average making the value of $[\mathrm{BLEU}(Y_r, Y_r^*) - U_r(\psi')]$ close to zero). In contrast to the conventional discriminative training methods such as MPE/MWE that treats all errors equally, the reestimation formula of (32) takes into account the end-to-end translation (or understanding or voice search) performance when training the acoustic model. Hence, different ASR errors are treated differently in terms of their impact on the ultimate goal of the SCIP task. This style of training gives the possibility to automatically dismiss certain types of errors so long as they can be tolerated by the MT (or NLU or IR) subsystem. This helps to strike a more balanced tradeoff with other errors that would affect MT (or NLU or IR) more negatively.

```
1. Build the baseline system to initialize { $\mathbf{\Gamma}, \boldsymbol{w}$ }.
2. Decode an N-best list or a lattice for training corpus using
   the baseline system, compute quality measure $C(Y_r, Y_r^*)$.
3. Set $\mathbf{\Gamma}' = \mathbf{\Gamma}$, $\boldsymbol{w}' = \boldsymbol{w}$.
4. Train parameters in the feature functions
   a. Go through the training set.
      i. Compute $P_{\mathbf{\Gamma}'}(Y_r, H_r|X_r)$ and $U_r(\mathbf{\Gamma}')$ .
      ii. Accumulate statistics $\{\gamma\}$.
   b. Update: $\mathbf{\Gamma}' \rightarrow \mathbf{\Gamma}$.
5. Train feature weights: $\boldsymbol{w}' \rightarrow \boldsymbol{w}$ (MERT/ MIRA/ Perceptron).
6. Test { $\mathbf{\Gamma}, \boldsymbol{w}$ } on the validation set.
7. Go to step 3 until training converges.
8. Pick the best { $\mathbf{\Gamma}, \boldsymbol{w}$ } on the validation set.
```

Fig. 3. ***Summary of the end-to-end optimization procedure for training a complete SCIP system.***

C. Iterative Training Process for End-to-End SCIP System Optimization

We now put together the full end-to-end optimization procedure for training a complete SCIP system. Since the parameter sets $\mathbf{\Gamma}$ and $\boldsymbol{w}$ affect the training of each other, we train them iteratively. That is, at each iteration, we first fix $\boldsymbol{w}$ and update $\mathbf{\Gamma}$, and then we retrain $\boldsymbol{w}$ given the new $\mathbf{\Gamma}$. In order to track the training progress, a validation set is used to determine the stop point of training. At the end, $\mathbf{\Gamma}$ and $\boldsymbol{w}$ that give the best score on the validation set are selected as the final parameter set. Fig. 3 provides a summary of the entire training procedure. Note that steps 2 and 4 are parallelizable across multiple processors.

$$\mu_i = \frac{\sum_r \sum_{H_r} \sum_{Y_r} \sum_t \gamma_G(H_r, Y_r, r, i, t) x_t + U(\psi')\tau\mu_i^0 + D_i\mu_i'}{\sum_r \sum_{H_r} \sum_{Y_r} \sum_t \gamma_G(H_r, Y_r, r, i, t) + U(\psi')\tau + D_i} \qquad (32)$$

VI. EXPERIMENTS AND ANALYSIS

In this section, we review a set of works published in the literature, which either supported or actually implemented various aspects of the end-to-end and joint-optimization-based approach to the design and learning of SCIP systems presented so far in this paper. We focus our attention mainly on the experimental evidence and evaluations that demonstrated the feasibility and effectiveness of the approach.

A. Spoken Language Translation

The initial proposal of using translation evaluation metrics to train both ASR and MT parameters in an SLT system was due to [94], where a primitive implementation and experimental evaluation showed promising results. The SLT scoring or decision function was developed based on Bayesian analysis on the joint ASR and MT components. The analysis led to the decision variable, used in SLT decoding, as a function of acoustic scores, source language model scores, and translation scores. A discriminative learning technique was further developed based on the decision-feedback principle that jointly learns the parameters in the source language model (used in ASR) and the MT subsystem in the overall SLT system. The SLT evaluation experiments were conducted on the International Workshop on Spoken Language Translation (IWSLT) DIALOG 2010 database. The experimental results demonstrated the effectiveness of this approach. Compared with the baseline system that assumes no ASR to MT interaction and no ASR uncertainty, the improved SLT system raised the BLEU score by 2.3 points, about half coming from the use of a combined posterior score from both ASR and MT subsystems (while keeping the original separate ASR and MT subsystem training, but generating an *n*-best list of the ASR output and using it in the downstream MT) and the remaining half from the joint training of the two subsystems.

The optimization criterion used in [94] was the posterior probability of the target text given the source speech signal, and the gradient descent was used to carry out the optimization process. The posterior probability is not the direct SLT evaluation metric of BLEU and this shortcoming was overcome in the more recent work of [40] and [38], both of which directly took BLEU as the optimization objective, as we presented in Section IV. The gradient–descent method of optimization, which took as many as 50 iterations to converge in training, was also improved to the EBW-based technique, with one magnitude fewer iterations to run in training. The optimization frameworks in the work of both [40] and [38] are two special cases of the more general framework we presented in Sections IV and V. The evaluation experiments were conducted on two tasks: 1) an IWSLT 2011 benchmark task where the EBW-based optimization technique on MT produced the best Chinese-to-English translation result on translating TED talks; and 2) Europarl German-to-English MT task where the same EBW-based technique leads to 1.1 BLEU point improvement over the state-of-the-art baseline system.

B. Spoken Language Understanding

While most of the SLU methods, which are reviewed in the recent book [98] and in Section IV-B, have taken the "divide and conquer" approach that separates the ASR "front–end" and the NLU "back–end," we draw attention to some notable exceptions here. In [86], Wang *et al.* questioned the conventional wisdom that better speech recognition accuracy is a good indicator for better SLU accuracy. Experimental evidence was provided that higher WERs may correlate with better slot filling accuracy as long as model training criteria match the optimization objective for understanding. Specifically, the experiments were conducted in the ATIS domain of SLU using the composite HMM/CFG. The use of domain knowledge and grammar library in the language model produced a higher WER (7.6%) in ASR than the use of a trigram language model trained with more data (WER of 6.0%), but slot filling understanding error rate is lower (8.8% versus 9.0%). A similar kind of divergence between the intermediate ASR word errors and the ultimate understanding errors was also found in earlier work of [74] and [25].

In a more recent work on the ATIS intent determination task, a decision-feedback learning method using a quantity correlated with intent classification accuracy was successfully applied to learn both the language model of the ASR component and the maximum-entropy model of the text classification component in the overall SLU system [89]. The jointly trained system produced some ASR errors but it performed better than the system assuming no ASR errors. The framework presented in Sections IV and V more systematically explores joint training of the system components. The objective functions are also more directly correlated with the performance metrics, which are applied not only to SLU but also to SLT, voice search, and other SCIP systems. Moreover, the optimization techniques are more principled and more general.

C. Voice Search

Like other SCIP systems, most of the existing voice search methods discard the uncertainty in the ASR output and the interactions between the ASR and IR subsystems. One main exception is the very recent work of [59], where an end-to-end ASR accuracy metric was proposed for voice search tasks, in the same spirit as the end-to-end performance metrics were developed for SLT and SLU. The end-to-end metric was motivated by the end user's experience and is intended to capture how this experience is affected by various ASR errors.

In the experiments reported in [59], it was shown that the impact of many types of ASR errors on the voice search quality is significantly smaller than what shows as the sentence error rates. That is, the voice search quality and ASR errors often are not well correlated. Such experimental observations offer a strong support to the basic premise of the end-to-end joint training of the voice search system as well as other SCIP systems that we have discussed in this paper.

D. Cross-Lingual Spoken Language Understanding

Porting an SLU service from one language to another is of tremendous practical value and hence of increasing interest to both the language understanding and machine translation research communities recently [57], [104]–[106]. One approach to addressing this need is to first translate the (testing) utterances in the second language to the primary language and then to use the primary languages SLU models to analyze them [104]. An alternative approach is to first translate the annotated (training) corpora to the second language, which is costly, followed by training models from understanding examples in the second language. Given the machine translation services that are broadly available nowadays, the SLU service for the primary language can be efficiently extended to cover a variety of other languages with minimal cost using the first approach. However, a full cross-lingual SLU system consists of multiple components including ASR, MT, and SLU. Due to the errors introduced in each of the components, the performance of straightforward cross-lingual SLU in this approach is far from acceptable. To address this issue, the framework presented in this paper has provided a principal solution to jointly train all the components to achieve an optimal end-to-end performance. Research along this direction is currently under way by the authors and their colleagues.

VII. SUMMARY AND FUTURE DIRECTIONS

In this paper, we organize and analyze a broad class of SCIP applications in the realm of human language technology. These include:

- SLT = ASR + MT;
- SLU = ASR + NLU;
- voice search = ASR + IR;
- cross-lingual SLU = ASR + MT + NLU;
- cross-lingual voice search = ASR + MT + IR;
- spoken dialog (open loop) = ASR + NLU + DialogControl;
- speech–speech translation = ASR + MT + SpeechSynthesis;

which are all enabled by a common component or subsystem of ASR that is in tandem with one or more downstream, text-based processing component(s). An overview of the work in the literature on SLT, SLU, voice search, and selected other SCIP systems is provided, setting up the background for a critique of the basic methodology in the current design of most of such systems.

Special challenges are examined for optimal construction of such complex information processing systems with the error-prone ASR component. Two distinct types of optimization inconsistency are analyzed: 1) mismatch between the training and deployment conditions; and 2) deviation of the training objective from the evaluation metric of the full system pipeline.

Aiming to overcome the optimization inconsistency, we establish a unified statistical framework applicable to all types of SCIP systems. We focus our technical presentation on two key aspects of the framework: 1) optimization objectives; and 2) optimization techniques. We also review a body of the work in the recent literature that implemented a number of isolated aspects of this general framework and demonstrated its feasibility and effectiveness. While most previous work on SCIP has focused on joint decoding with the parameters of component models being trained disjointly (i.e., without considering their interactions), we emphasize in this paper joint optimization for the full set of parameters in the overall SCIP system.

As is clear from the reviews and presentation conducted in this paper, SCIP systems are complex with difficult optimization problems in their design and learning. While some progress has been made, many challenges remain. First, prosody is an important aspect of the speech signal, interacting with both ASR and its downstream components strongly. How to optimally embed prosody in the framework presented in this paper [e.g., designing tightly coupled features $\varphi_i(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})$ in (10)] has by no means an obvious solution. Second, how to acquire a large amount of supervised training data for optimizing SCIP systems is more difficult than that for optimizing separate ASR or text-based MT, NLU, and IR systems. While, in principle, end-to-end two-way parallel data are sufficient for the full SCIP system training, practical difficulties of associating the end-to-end data and labels may necessitate three-way parallel data collection, which is very costly. In practice, it may be feasible to first train the individual components separately, then to apply the end-to-end joint optimization approach to fine-tune the models. Using proper regularization described in this paper, the end-to-end training can be achieved with a small amount of three-way parallel data. Third, in practical usage scenarios of SCIP systems, users may have the desire not only for having the final system's output but also for observing some intermediate results. For example, in the SLT system, it is desirable to show the end users, who are often ignorant of the target language, not just high-quality translated target language but also the ASR results on the source language with reasonably low ASR error rates. To fulfill such desire, the objective function in end-to-end learning may be more complicated than described in Section IV. Fourth, successful exploration and exploitation of the equivalence of generative and log-linear discriminative models [43] has the potential to further extend the feasibility of current EBW-based learning strategy to attack more challenging problems in the SCIP system design and optimization. Fifth, despite the best effort to overcome it, there still remains some degree of inconsistency between the training objective, as exemplified in (3) and (19), and the decision variable in decoding, as exemplified in (5). Integration of a minimum Bayes-risk decoding framework [31] into the current end-to-end optimization strategy holds promise to eliminate this final piece of inconsistency. Finally, in light of the recent advance in deep learning methods that have dramatically cut down the ASR error rate [17], [44], [92], it is highly desirable to extend the end-to-end optimization approach for the SCIP systems presented in this paper based on Gaussian-HMM ASR subsystems to incorporate the potentially new generation of ASR based on deep networks. ■

APPENDIX I
DERIVATION STEPS FOR (23)

Here we show the derivation steps leading to the rational-function form of (23) for the transformed training objective function $R(\theta)$.

Substituting (10) into (12) and then into (1), we obtain

$$U(\boldsymbol{\theta}) = \frac{1}{Z}\sum_{\boldsymbol{Y}}\sum_{\boldsymbol{H}} \exp\left\{\sum_i w_i \log \varphi_i(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})\right\} C(\boldsymbol{Y}) \tag{35}$$

where $Z = \Sigma_Y \Sigma_H \exp\{\Sigma_i w_i \log \varphi_i(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})\}$.

Further algebraic manipulations yield

$$\begin{aligned} U(\boldsymbol{\theta}) &= \frac{\sum_{\boldsymbol{Y}}\sum_{\boldsymbol{H}} \exp\left\{\log\prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})\right\} C(\boldsymbol{Y})}{\sum_{\boldsymbol{Y}}\sum_{\boldsymbol{H}} \exp\left\{\log\prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})\right\}} \\ &= \frac{\sum_{\boldsymbol{Y}}\sum_{\boldsymbol{H}} \prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X}) C(\boldsymbol{Y})}{\sum_{\boldsymbol{Y}}\sum_{\boldsymbol{H}} \prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})}. \end{aligned} \tag{36}$$

On the other hand, we rewrite (17) to obtain

$$\mathrm{KL}(\boldsymbol{\theta}^0\|\boldsymbol{\theta}) = -\sum_i\sum_j \theta_{ij}^0 \log\theta_{ij} + C \tag{37}$$

where C is a term irrelevant to optimizing $\boldsymbol{\theta}$. Equation (37) can be further written into

$$e^{-\tau\cdot\mathrm{KL}(\boldsymbol{\theta}^0\|\boldsymbol{\theta})} = \prod_i\prod_j \theta_{ij}^{\tau\theta_{ij}^0}. \tag{38}$$

After substituting (36) and (38) into (22), we obtain (23).

APPENDIX II
DERIVATION STEPS FOR (25)

Here we provide the derivation steps leading to the EBW reestimation formula in (25) for the phrase translation model parameters in an SLT system.

We start from the transformed objective function $R(\boldsymbol{\theta})$ in (23), follow the first step of the EBW algorithm described in Section V-B2, and construct the following auxiliary function:

$$F(\boldsymbol{\theta};\boldsymbol{\theta}') = G(\boldsymbol{\theta})\cdot J(\boldsymbol{\theta}) - H(\boldsymbol{\theta})R(\boldsymbol{\theta}').$$

Noting only $\boldsymbol{\theta}$, not $\boldsymbol{\theta}'$, contains the parameters p_{ij} for optimization, we obtain

$$\frac{\partial F(\boldsymbol{\theta};\boldsymbol{\theta}')}{\partial p_{ij}} = \frac{\partial G(\boldsymbol{\theta})}{\partial p_{ij}}J(\boldsymbol{\theta}) + \frac{\partial J(\boldsymbol{\theta})}{\partial p_{ij}}G(\boldsymbol{\theta}) - R(\boldsymbol{\theta}')\frac{\partial H(\boldsymbol{\theta})}{\partial p_{ij}}$$

where, according to (23), we have

$$\begin{aligned} \frac{\partial G(\boldsymbol{\theta})}{\partial p_{ij}} &= \sum_Y\sum_H\prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})C(\boldsymbol{Y})\frac{\partial \log\varphi_{\mathrm{FP}}^{w_{\mathrm{FP}}}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})}{\partial p_{ij}} \\ &= w_{\mathrm{FP}}\frac{1}{p_{ij}}\cdot\sum_Y\sum_H\prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})C(\boldsymbol{Y}) \\ &\quad\cdot\sum_{r,k}\mathbf{1}(\bar{h}_{r,k}=i,\bar{y}_{r,k}=j) \\ \frac{\partial J(\boldsymbol{\theta})}{\partial p_{ij}} &= J(\boldsymbol{\theta})\frac{\partial \log J(\boldsymbol{\theta})}{\partial p_{ij}} \\ &= J(\boldsymbol{\theta})\frac{\partial \sum_i\sum_j \tau\theta_{ij}^0\log\theta_{ij}}{\partial p_{ij}} \\ &= J(\boldsymbol{\theta})\tau p_{ij}^0\frac{1}{p_{ij}} \\ \frac{\partial H(\boldsymbol{\theta})}{\partial p_{ij}} &= w_{\mathrm{FP}}\frac{1}{p_{ij}}\cdot\sum_Y\sum_H\prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X}) \\ &\quad\cdot\sum_{r,k}\mathbf{1}(\bar{h}_{r,k}=i,\bar{y}_{r,k}=j). \end{aligned}$$

Then

$$\begin{aligned} \left.\frac{\partial F(\boldsymbol{\theta};\boldsymbol{\theta}')}{\partial p_{ij}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}'} &= J(\boldsymbol{\theta}')w_{\mathrm{FP}}\frac{1}{p'_{ij}}H(\boldsymbol{\theta}') \\ &\quad\cdot\sum_Y\sum_H\gamma_{\mathrm{FP}}(\boldsymbol{H},\boldsymbol{Y},i,j) + G(\boldsymbol{\theta}')J(\boldsymbol{\theta}')\tau p_{ij}^0\frac{1}{p'_{ij}} \end{aligned}$$

$$p_{ij} = \frac{w_{\mathrm{FP}} J(\theta')H(\theta') \sum_Y \sum_H \gamma_{\mathrm{FP}}(H,Y,i,j) + G(\theta')J(\theta')\tau p^0_{ij} + Dp'_{ij}}{\sum_j w_{\mathrm{FP}} J(\theta')H(\theta') \sum_Y \sum_H \gamma_{\mathrm{FP}}(H,Y,i,j) + G(\theta')J(\theta')\tau + D}$$

$$= \frac{\sum_Y \sum_H \gamma_{\mathrm{FP}}(H,Y,i,j) + \dfrac{G(\theta')}{H(\theta')}\dfrac{\tau}{w_{\mathrm{FP}}} p^0_{ij} + \dfrac{D}{w_{\mathrm{FP}} J(\theta')H(\theta')} p'_{ij}}{\sum_j \sum_Y \sum_H \gamma_{\mathrm{FP}}(H,Y,i,j) + \dfrac{G(\theta')}{H(\theta')}\dfrac{\tau}{w_{\mathrm{FP}}} + \dfrac{D}{w_{\mathrm{FP}} J(\theta')H(\theta')}}$$

where

$$\gamma_{\mathrm{FP}}(\boldsymbol{H},\boldsymbol{Y},i,j) = \frac{\prod_i \varphi_i^{w_i}(\boldsymbol{Y},\boldsymbol{H},\boldsymbol{X})|_{\boldsymbol{\theta}=\boldsymbol{\theta}'}}{H(\boldsymbol{\theta}')} \cdot \left[C(\boldsymbol{Y}) - \frac{G(\boldsymbol{\theta}')}{H(\boldsymbol{\theta}')}\right] \cdot \sum_{r,k} \mathbf{1}(\bar{h}_{r,k} = i, \bar{y}_{r,k} = j)$$
$$= P_{\boldsymbol{\theta}'}(\boldsymbol{Y},\boldsymbol{H}|\boldsymbol{X}) \cdot [C(\boldsymbol{Y}) - U(\boldsymbol{\theta}')] \cdot \sum_{r,k} \mathbf{1}(\bar{h}_{r,k} = i, \bar{y}_{r,k} = j).$$

Substituting the above equation into the EBW formula of (21), we obtain the equation shown at the top of the page.

Note that $D/w_{\mathrm{FP}}J(\theta')H(\boldsymbol{\theta}')$ is independent from $\boldsymbol{\theta}$ so we denote this ratio as D without loss of generality. Similarly, we denote by τ/w_{FP} as τ_{FP}. Further, using $G(\boldsymbol{\theta}')/H(\boldsymbol{\theta}') = U(\boldsymbol{\theta}')$, we obtain

$$p_{ij} = \frac{\sum_Y \sum_H \gamma_{\mathrm{FP}}(\boldsymbol{H},\boldsymbol{Y},i,j) + U(\boldsymbol{\theta}')\tau_{\mathrm{FP}} p^0_{ij} + Dp'_{ij}}{\sum_j \sum_Y \sum_H \gamma_{\mathrm{FP}}(\boldsymbol{H},\boldsymbol{Y},i,j) + U(\boldsymbol{\theta}')\tau_{\mathrm{FP}} + D}.$$

The right-hand side of the above equation can be further rewritten into the decomposed form of (25) following the derivation steps detailed in [41].

REFERENCES

[1] M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. Strope, "Deploying GOOG-411: Early lessons in data, measurement, and testing," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2008, pp. 5260–5263.

[2] A. Bagnell and D. Bradley, "Differentiable sparse coding," *Proc. Neural Inf. Process. Syst.*, pp. 113–120, 2008.

[3] J. Baker, L. Deng, J. Glass, S. Khudanpur, C.-H. Lee, N. Morgan, and D. O'Shaughnessy, "Research developments and directions in speech recognition and understanding—Part 1," *IEEE Signal Process. Mag.*, vol. 26, no. 3, pp. 75–80, May 2009.

[4] J. Baker, L. Deng, S. Khudanpur, C. Lee, J. Glass, N. Morgan, and D. O'Shaughnessy, "Updated MINDS report on speech recognition and understanding—Part 2," *IEEE Signal Process. Mag.*, vol. 26, no. 4, pp. 78–85, Jul. 2009.

[5] R. Battiti, "First- and second- order methods for learning: Between steepest descent and Newton's method," *Neural Comput.*, vol. 4, pp. 141–166, 1992.

[6] L. Baum and G. Sell, "Growth transformations for functions on manifolds," *Pacific J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.

[7] L. Baum and J. Eagon, "An inequality with applications to statistical prediction for functions of Markov processes and to a model of ecology," *Bull. Amer. Math. Soc.*, vol. 73, pp. 360–363, 1967.

[8] N. Bertoldi, R. Zens, M. Federico, and W. Shen, "Efficient speech translation through confusion network decoding," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 8, pp. 1696–1705, Nov. 2008.

[9] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.

[10] D. Bohus, S. G. Puerto, D. Huggins-Daines, V. Keri, G. Krishna, R. Kumar, A. Raux, and S. Tomkoohus, "ConQuest: An open-source dialog system for conferences," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., Human Lang. Technol.*, 2007, pp. 9–12.

[11] D. Bolanos, G. Zweig, and P. Nguyen, "Multi-scale personalization for voice search applications," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., Human Lang. Technol.*, 2009, pp. 101–104.

[12] R. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[13] P. Brown, S. Pietra, V. Pietra, and R. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, 1993.

[14] F. Casacuberta, M. Federico, H. Ney, and E. Vidal, "Recent efforts in spoken language translation," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 80–88, May 2008.

[15] C. Chelba, T. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 39–49, May 2008.

[16] S. Axelrod, V. Goel, R. Gopinath, P. Olsen, and K. Visweswariah, "Discriminative estimation of subspace constrained Gaussian mixture models for speech recognition," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 1, pp. 172–189, Jan. 2007.

[17] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[18] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur, "Spoken language understanding," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 50–58, May 2008.

[19] L. Deng and D. O'Shaughnessy, *Speech Processing—A Dynamic and Optimization-Oriented Approach*. New York: Marcel Dekker, 2003.

[20] L. Deng and X. Huang, "Challenges in adopting speech recognition," *Commun. ACM*, vol. 47, no. 1, pp. 11–13, Jan. 2004.

[21] L. Deng, K. Wang, A. Acero, H. Hon, J. Droppo, C. Boulis, Y. Wang, D. Jacoby, M. Mahajan, C. Chelba, and X. D. Huang, "Distributed speech processing in MiPad's multimodal user interface," *IEEE Trans. Audio Speech Process.*, vol. 10, no. 8, pp. 605–619, Nov. 2002.

[22] L. Deng, "Front-end, back-end, and hybrid techniques to noise-robust speech recognition," in *Robust Speech Recognition of Uncertain Data*, D. Kolossa and R. Haeb-Umbach, Eds. New York: Springer-Verlag, 2011, pp. 67–99.

[23] J. E. Dennis and R. B. Schnabel, "Numerical methods for unconstrained optimization and nonlinear equations *SIAM's Classics in Applied Mathematics*. Philadelphia, PA: SIAM, 1996.

[24] J. Droppo and A. Acero, "Joint discriminative front end and back end training for improved speech recognition accuracy," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2006, DOI: 10.1109/ICASSP.2006.1660012.

[25] Y. Estève, C. Raymond, F. Bechet, and R. DeMori, "Conceptual decoding for spoken dialog systems," in *Proc. Eurospeech Conf.*, Geneva, Switzerland, Sep. 1–4, 2003.

[26] S. E. Fahlman, "An empirical study of learning speed in back-propagation

networks," Canergie Mellon Univ., Pittsburgh, PA, Tech. Rep., 1988.

[27] J. Feng, S. Bangalore, and M. Gilbert, "Role of natural language understanding in voice local search," in *Proc. Interspeech 2009*, Brighton, U.K., Sep. 6–10, 2009.

[28] J. Feng, "Query parsing in mobile voice search," in *Proc. World Wide Web 2010*, Raleigh, NC, USA, Apr. 26–30, 2010.

[29] J. Feng, M. Johnston, and S. Bangalore, "Speech and multimodal interaction in mobile search," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 40–49, Jul. 2011.

[30] A. Franz and B. Milch, "Searching the Web by voice," in *Proc. Conf. Comput. Linguist.*, 2002, pp. 1213–1217.

[31] V. Goel, S. Kumar, and W. Byrne, "Segmental minimum Bayes-risk decoding for automatic speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 3, pp. 234–249, May 2004.

[32] P. Gopalakrishnan, D. Kanevsky, A. Nadas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 107–113, Jan. 1991.

[33] A. Gunawardana and W. Byrne, "Discriminative speaker adaptation with conditional maximum likelihood linear regression," in *Proc. Eurospeech 2001*, Aalborg, Denmark, Sep. 3–7, 2001.

[34] S. Hahn, M. Dinarelli, C. Raymond, F. Lefèvre, P. Lehnen, R. De Mori, H. Ney, and G. Riccardi, "Comparing stochastic approaches to spoken language understanding in multiple languages," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 6, pp. 1569–1583, Aug. 2011.

[35] D. Harman, *Meeting of the MINDS: Future Directions for Human Language Technology*. [Online]. Available: http://www-nlpir.nist.gov/MINDS/FINAL/exec.summary.pdf

[36] K. Hashimoto, J. Yamagishi, W. Byrne, S. King, and K. Tokuda, "Impacts of machine translation and speech synthesis on speech-to-speech translation," *Speech Commun.*, vol. 54, no. 7, pp. 857–866, Sep. 2012.

[37] X. He and L. Deng, "Speech recognition, machine translation, and speech translation—A unified discriminative learning paradigm," *IEEE Signal Process. Mag.*, vol. 28, no. 5, pp. 126–133, Sep. 2011.

[38] X. He and L. Deng, "Maximum expected BLEU training of phrase and lexicon translation models," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, Jul. 2012, vol. 1, pp. 292–301.

[39] X. He, L. Deng, and A. Acero, "Why word error rate is not a good metric for speech recognizer training for the speech translation task?" in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2011, pp. 5632–5635.

[40] X. He, A. Axelrod, L. Deng, A. Acero, M. Hwang, A. Nguyen, A. Wang, and X. Huang, "The MSR system for IWSLT 2011 evaluation," in *Proc. IWSLT*, San Francisco, CA, USA, Dec. 8–9, 2011.

[41] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 14–36, Sep. 2008.

[42] G. Heigold, P. Dreuw, S. Hahn, R. Schlüter, and H. Ney, "Margin-based discriminative training for string recognition," *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 6, pp. 917–925, Dec. 2010.

[43] G. Heigold, H. Ney, P. Lehnen, T. Gass, and R. Schluter, "Equivalence of generative and log-linear models," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 5, pp. 1138–1148, Jul. 2011.

[44] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[45] R. Hsiao and T. Schultz, "Generalized Baum-Welch algorithm and its implication to a new extended Baum-Welch algorithm," in *Proc. Interspeech 2011*, Florence, Italy, Aug. 27–31, 2011.

[46] X. Huang and L. Deng, "An overview of modern speech recognition," in *Handbook of Natural Language Processing*, 2nd ed. London, U.K.: Chapman & Hall/CRC Press, 2010, pp. 339–366.

[47] K. Jarvelin and J. Kekalainen, "IR evaluation methods for retrieving highly relevant documents," in *Proc. Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2000, pp. 41–48.

[48] Y. Ju and T. Paek, "A voice search approach to replying to SMS messages in automobiles," in *Proc. Interspeech 2009*, Brighton, U.K., Sep. 6–10, 2009.

[49] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 2008.

[50] D. Kanevsky, D. Nahamoo, T. Sainath, B. Ramabhadran, and P. Olsen, "A-functions: A generalization of extended Baum-Welch transformations to convex optimization," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2011, pp. 5164–5167.

[51] D. Kanevsky, D. Nahamoo, T. Sainath, and B. Ramabhadran, "Convergence of line search A-function methods," in *Proc. Interspeech 2011*, Florence, Italy, Aug. 27–31, 2011.

[52] P. Koehn, F. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., Human Lang. Technol.*, 2003, vol. 1, pp. 48–54.

[53] D. Kolossa and R. Haeb-Umbach, Eds., *Robust Speech Recognition of Uncertain Data*. New York: Springer-Verlag, 2011.

[54] S. Krauwer, D. Arnold, W. Kasper, M. Rayner, and H. Somers, "Spoken language translation," in *Proc. ACL Spoken Lang. Transl. Workshop*, Madrid, Spain, 1997, pp. 1–5.

[55] J. Le Roux and E. McDermott, "Optimization for discriminative training," in *Proc. Interspeech 2005*, Lisbon, Portugal, Sep. 4–8, 2005.

[56] M. Lehr and I. Shafran, "Discriminatively estimated joint acoustic, duration and language model for speech recognition," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2010, pp. 5542–5545.

[57] F. Lefèvre, F. Mairesse, and S. Young, "Cross-lingual spoken language understanding from unaligned data using discriminative classification models and machine translation," in *Proc. Interspeech 2010*, Makuhari, Japan, Sep. 26–30, 2010.

[58] L.-S. Lee and B. Chen, "Spoken document understanding and organization," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 42–60, Sep. 2005.

[59] M. Levit, S. Chang, B. Buntschuh, and N. Kibre, "End-to-end speech recognition accuracy metric for voice search tasks," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 5141–5144.

[60] X. Li, Y. Wang, and G. Tur, "Multi-task learning for spoken language understanding with shared slots," in *Proc. Interspeech 2011*, Florence, Italy, Aug. 27–31, 2011.

[61] S. Mann, A. Berton, and U. Ehrlich, "How to access audio files of large databases using in-car speech dialogue systems," in *Proc. Interspeech Conf.*, Antwerp, Belgium, 2007, pp. 138–141.

[62] E. Matusov, S. Kanthak, and H. Ney, "Integrating speech recognition and machine translation: Where do we stand?" in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2006, DOI: 10.1109/ICASSP.2006.1661501.

[63] E. McDermott and T. Hazen, "Minimum classification error training of landmark models for real-time continuous speech recognition," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2006, vol. 1, pp. 937–940.

[64] E. McDermott, T. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large vocabulary speech recognition using minimum classification error," *IEEE Trans. Speech Audio Process.*, vol. 15, no. 1, pp. 203–223, Jan. 2007.

[65] S. Nakamura, K. Markov, H. Nakaiwa, G. Kikui, H. Kawai, T. Jitsuhiro, J. Zhang, H. Yamamoto, E. Sumita, and S. Yamamoto, "The ATR multilingual speech-to-speech translation system," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 2, pp. 365–376, Mar. 2006.

[66] H. Ney, "Speech translation: Coupling of recognition and translation," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 1999, vol. 1, pp. 517–520.

[67] F. Och, "Minimum error rate training in statistical machine translation," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2003, pp. 160–167.

[68] J. Olive, C. Christianson, and J. McCary, Eds., *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*. New York: Springer-Verlag, 2011.

[69] M. Ostendorf, B. Favre, R. Grishman, D. Hakkani-Tur, M. Harper, D. Hillard, J. Hirschberg, J. Heng, J. Kahn, L. Yang, S. Maskey, E. Matusov, H. Ney, A. Rosenberg, E. Shriberg, W. Wen, and C. Woofers, "Speech segmentation and spoken document processing," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 59–69, May 2008.

[70] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2002, pp. 311–318.

[71] M. Paul, M. Federico, and S. Stücker, "Overview of the IWSLT 2010 evaluation campaign," in *Proc. IWSLT*, Paris, France, Dec. 2–3, 2010.

[72] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge Univ. Eng. Dept., Cambridge Univ., Cambridge, U.K., 2004.

[73] A. Reddy and R. Rose, "Integration of statistical models for dictation of document translations in a machine-aided human translation task," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 8, pp. 2015–2027, Nov. 2010.

[74] G. Riccardi and A. L. Gorin, "Stochastic language models for speech recognition and understanding," in *Proc. ICSLP*, Sydney, Australia, Nov. 30–Dec. 4, 1998.

[75] M. Riedmiller and H. Braun, "A direct adaptive method for faster back propagation learning: The RPROP algorithm," in *Proc. IEEE Int. Conf. Neural Netw.*, San Francisco, CA, 1993, pp. 586–591.

[76] Y. Normandin, "Hidden Markov models, maximum mutual information estimation, and the speech recognition problem," Ph.D. dissertation, Electr. Comput. Eng. Dept., McGill Univ., Montreal, QC, Canada, 1991.

[77] M. Seltzer, Y. Ju, I. Tashev, Y. Wang, and D. Yu, "In-car media search," *IEEE Signal Process. Mag.*, vol. 28, no. 4, pp. 50–60, Jul. 2011.

[78] P. Simard, Y. Steinkraus, and J. Platt, "Best practice for convolutional neural networks applied to visual document analysis," in *Proc. Int. Conf. Document Anal. Recognit.*, 2003, pp. 958–962.

[79] Y. Song, Y.-Y. Wang, Y. Ju, M. Seltzer, I. Tashev, and A. Acero, "Voice search of structured media data," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2009, pp. 3941–3944.

[80] G. Tur and L. Deng, "Intent determination and spoken utterance classification," in *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, Tur and De Mori, Eds. New York: Wiley, 2011, pp. 81–104.

[81] G. Tur, L. Deng, D. Hakkani-Tür, and X. He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 5045–5048.

[82] E. Vidal, "Finite-state speech-to-speech translation," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Munich, Germany, 1997, pp. 111–114.

[83] A. Waibel and C. Fugen, "Spoken language translation," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 70–79, May 2008.

[84] Y. Wang, D. Yu, Y. Ju, and A. Acero, "An introduction to voice search," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 28–38, May 2008.

[85] Y. Wang, L. Deng, and A. Acero, "Semantic frame based spoken language understanding," in *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, Tur and De Mori, Eds. New York: Wiley, 2011, pp. 35–80.

[86] Y. Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, 2003, pp. 577–582.

[87] Y. Wang, L. Deng, and A. Acero, "Spoken language understanding," *IEEE Signal Process. Mag.*, vol. 22, no. 5, pp. 16–31, Sep. 2005.

[88] P. Xu, P. Fung, and R. Chan, "Phrase-level transduction model with reordering for spoken to written language transformation," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 4965–4968.

[89] S. Yaman, L. Deng, D. Yu, Y. Wang, and A. Acero, "An integrative and discriminative technique for spoken utterance classification," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 6, pp. 1207–1214, Aug. 2008.

[90] S. Young, "Cognitive user interfaces," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 128–140, May 2010.

[91] S. Young, M. Gasic, B. Thomson, and J. Williams, "POMDP-based statistical spoken dialog systems: A review," *Proc. IEEE*, 2013, DOI: 10.1109/JPROC.2012.2225812.

[92] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 145–154, Jan. 2011.

[93] D. Yu, Y.-C. Ju, Y.-Y. Wang, G. Zweig, and A. Acero, "Automated directory assistance system: From theory to practice," in *Proc. Interspeech Conf.*, Antwerp, Belgium, 2007, pp. 2709–2712.

[94] Y. Zhang, L. Deng, X. He, and A. Acero, "A novel decision function and the associated decision-feedback learning for speech translation," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2011, pp. 5608–5611.

[95] B. Zhou, "Statistical translation for speech: A perspective on structures and learning," *Proc. IEEE*, 2013.

[96] B. Zhou, L. Besacier, and Y. Gao, "On efficient coupling of ASR and SMT for speech translation," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2007, vol. IV, pp. 101–104.

[97] G. Zweig, P. Nguyen, Y.-C. Ju, Y.-Y. Wang, D. Yu, and A. Acero, "The voice rate dialog system for consumer ratings," in *Proc. Interspeech Conf.*, 2007, pp. 2713–2716.

[98] Tur and De Mori, Eds., *Spoken Language Understanding: Systems for Extracting Semantic Information From Speech*. New York: Wiley, 2011.

[99] D. Chiang, K. Knight, and W. Wang, "11,001 new features for statistical machine translation," in *Proc. of Conf. North Amer. Chapter Assoc. Comput. Linguist., Human Lang. Technol.*, 2009, pp. 218–226.

[100] P. Liang, A. Bouchard-Cote, D. Klein, and B. Taskar, "An end-to-end discriminative approach to machine translation," in *Proc. Conf. Comput. Linguist./Assoc. Comput. Linguist.*, 2006, pp. 761–768.

[101] B. Jabaian, L. Besacier, and F. Lefevre, "Comparison and combination of lightly supervised approaches for language portability of a spoken language understanding system," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 3, 2013.

[102] L. Deng, A. Acero, M. Plumpe, and X. D. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," *Proc. Int. Conf. Spoken Lang.*, pp. 806–809.

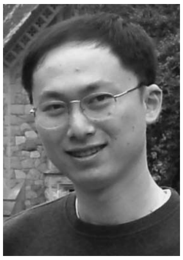
[103] L. Deng, G. Tur, X. He, and D. Hakkani-Tur, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," *Proc. IEEE Workshop Spoken Lang. Technol.*, Dec. 2012.

[104] B. Jabaian, L. Besacier, and F. Lefevre, "Investigating multiple approaches for SLU portability to a new language," in *Proc. Interspeech*, 2010.

[105] B. Jabaian, L. Besacier, and F. Lefevre, "Combination of stochastic understanding and machine translation systems for language portability of dialogue systems," in *Proc. ICASSP*, 2011.

[106] N. Camelin, C. Raymond, F. Bechet, and R. De Mori, "On the use of machine translation for spoken language understanding portability," in *Proc. ICASSP*, 2010.

ABOUT THE AUTHORS



Xiaodong He (Senior Member, IEEE) received the B.S. degree from Tsinghua University, Beijing China, in 1996, the M.S. degree from the Chinese Academy of Science, Beijing, China, in 1999, and the Ph.D. degree from the University of Missouri—Columbia, Columbia, USA, in 2003.

He is a Researcher at Microsoft Research, Redmond, WA, and an Affiliate Professor in the Department of Electrical Engineering, University of Washington, Seattle. His research interests include machine learning, speech recognition, translation, and understanding, machine translation, natural language processing, and information retrieval. He has published over 50 technical papers and one book in these areas. In benchmark evaluations, he and his colleagues have developed entries that obtained No. 1 place at the 2008 NIST Machine Translation Evaluation and the 2011 International Workshop on Spoken Language Translation Evaluation, both in Chinese/English translation, respectively.

Dr. He currently serves as an Associate Editor of the IEEE Signal Processing Magazine. He also served as the Guest Editor of the IEEE Transactions on Audio, Speech, and Language Processing and the IEEE Journal of Selected Topics in Signal Processing. He serves as Cochair of Special Sessions of the 2013 International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Cochair of the 2008 Neural Information Processing Systems (NIPS) Workshop on Speech and Language, and on program committees of major speech and language processing conferences. He is a member of the Association for Computational Linguistics (ACL).

Li Deng (Fellow, IEEE) received the Ph.D. degree from the University of Wisconsin—Madison, Madison.

He joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 1989, as an Assistant Professor, where he became a tenured full Professor in 1996. In 1999, he joined Microsoft Research, Redmond, WA, as a Senior Researcher, where he is currently a Principal Researcher. Since 2000, he has also been an Affiliate Full Professor and graduate committee member in the Department of Electrical Engineering, University of Washington. Prior to MSR, he also worked or taught at the Massachusetts Institute of Technology (Cambridge), ATR Interpreting Telecom. Research Laboratory (Kyoto, Japan), and the Hong Kong University of Science and Technology (HKUST). In the general areas of speech/language technology, machine learning, and signal processing, he has published over 300 refereed papers in leading journals and conferences and three books, and has given keynotes, tutorials, and distinguished lectures worldwide. His recent technical work (since 2009) on industry-scale deep learning with colleagues and collaborators has created significant impact on speech recognition, signal processing, and related applications.

Prof. Deng is a Fellow of the Acoustical Society of America and the International Speech Communication Association (ISCA). He served on the Board of Governors of the IEEE Signal Processing Society (2008–2010). More recently, he served as the Editor-in-Chief for the IEEE Signal Processing Magazine (2009–2011), which earned the highest impact factor among all IEEE publications and for which he received the 2011 IEEE SPS Meritorious Service Award. He currently serves as the Editor-in-Chief for the IEEE Transactions on Audio, Speech, and Language Processing.