ADVERSARIAL BANDIT FOR ONLINE INTERACTIVE ACTIVE LEARNING OF ZERO-SHOT SPOKEN LANGUAGE UNDERSTANDING

Emmanuel Ferreira, Alexandre Reiffers Masson, Bassam Jabaian and Fabrice Lefèvre

CERI-LIA, University of Avignon, France

firstname.lastname@univ-avignon.fr

ABSTRACT

Many state-of-the-art solutions for the understanding of speech data have in common to be probabilistic and to rely on machine learning algorithms to train their models from large amount of data. The difficulty remains in the cost of collecting and annotating such data. Another point is the time for updating an existing model to a new domain. Recent works showed that a zero-shot learning method allows to bootstrap a model with good initial performance. To do so, this method relies on exploiting both a small-sized ontological description of the target domain and a generic word-embedding semantic space for generalization. Then, this framework has been extended to exploit user feedbacks to refine the zero-shot semantic parser parameters and increase its performance online. In this paper, we propose to drive this online adaptive process with a policy learnt using the Adversarial Bandit algorithm Exp3. We show, on the second Dialog State Tracking Challenge (DSTC2) datasets, that this proposition can optimally balance the cost of gathering valuable user feedbacks and the overall performance of the spoken language understanding module.

Index Terms— Spoken language understanding, zero-shot learning, bandit problem, out-of-domain training data, online adaptation.

1. INTRODUCTION

In a dialogue system, the Spoken Language Understanding (SLU) module is dedicated to extract for each incoming user utterance some hypotheses about its semantic content (e.g. meaning). Generally, the latter is expressed as a sequence of Dialogue Acts (DAs) in the form `acttype(slot=value)`. The `acttype`s are task-independent and convey the user intent behind its communicative act, while `slot`s and `value`s are domain dependent and correspond to the specific pieces of information that the system can manipulate (e.g. entries in a back-end database, commands to a robot/device). For instance, the utterance "hello i am looking for a french restaurant in the south part of town" corresponds to the dialogue act sequence "`hello(), inform(food=french), inform(area=south)`".

State-of-the-art SLU systems are based on probabilistic approaches trained on a large amount of data with various machine learning methods (see for instance [1, 2, 3, 4]). Training corpus size has an important influence on the quality of the system. But collecting such a corpus, despite consequent efforts to reduce it such as in [5], is costly in time and human expertise. Most of the time dedicated to create a dialogue system is for the data collection and annotation [6].

Some research works have focused on the use of lightly supervised [7, 8, 9], or unsupervised [10, 11] training approaches to cope with the lack of annotated resources by either exploiting the semantic web for mining additional training data and enriching classification features or proposing unsupervised annotation process on a close-domain corpus [5, 12]. With the same objective of minimising the cost of data collection, some other works focused on porting a system across language and domain [13, 14]. Many propositions consider an Active Learning (AL) procedure to reduce corpus annotation and verification time [15, 16, 17], or to build a mini corpus to bootstrap a first system further used in an iterative data collection process, as in [18]. In [19] an instance-based approach for online adaptation of semantic models is presented, while [20] proposes a supervised approach for updating the SLU models with a limited supervision given by users calling the system.

In [21, 22] a zero-shot learning method for SLU, the Zero-Shot Semantic Parser (ZSSP), based on word embeddings (word2vec [23]) is applied to generalize an initial knowledge base about the task at hand (e.g. database entries, partial ontological description). This approach requires neither annotated data nor in-context data and can reach instantly state-of-the-art performance. Along this line, an online adaptive strategy was also proposed to refine the model in an incremental fashion with a light supervision. Indeed, with this strategy, the users were only asked to confirm some hypotheses made by the system (yes/no questions) but not to explicitly correct any error. As such it allows them to correct some classification errors but never to add new concepts or values in the model and thus does not allow any domain extension. In this paper, we propose to extend this online adaptation strategy in order to also address this issue and thus be able to expand the model with new knowledge continuously.

To define this new strategy we propose to cast the online AL problem of the understanding module into an Adversarial Bandit one. This setting aims to minimize the supervision costs while asking the user questions with the maximum impact on the model. Bandit algorithms have been widely studied in the machine learning community [24, 25] with the objective to obtain the best solutions to the exploration-exploitation dilemma. They choose between exploiting options that yielded the best output (payoffs) in previous iterations and exploring new options that might give higher performance in the future. Only few works have already employed this kind of methods to optimise a vocal interaction system. For instance [26] applied a multiclass Bandit algorithms to train a call-type classifier with yes/no user feedbacks. We show that our proposed technique allows to achieve good performance with a low and adjustable supervision cost on the second Dialog State Tracking (DSTC2) testbed [27].

In Section 2 the basis of our baseline ZSSP SLU model are recalled. Section 3 describes the proposed bandit-based strategy for the online adaptation employed for its refinement. Then, the experimental study is presented in Section 4 followed by some concluding remarks and perspectives.

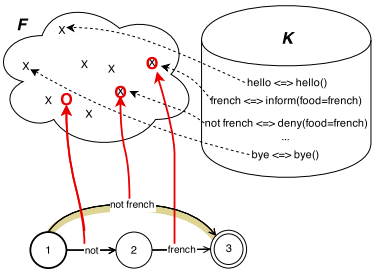


Fig. 1: ZSSP model

2. ZERO-SHOT LEARNING MODEL FOR SPOKEN LANGUAGE UNDERSTANDING

The SLU model employed in our study is the ZSSP zero-shot learning model proposed in [21, 22]. The latter, depicted in Figure 1, makes use of three main components. The first one is a semantic feature space F based on word embeddings learnt with neural network algorithms on open domain data [23, 28].

The second component is a semantic knowledge base K which contains some examples of lexical segments or phrases (called chunks hereafter) associated to each targeted DA (as inferred from the ontology and database). In K, assignment coefficients measure the correspondence between a chunk and any known DAs. Indeed, these values, which are in range $[0, 1]$, denote how confident the model is about the fact that a chunk is related to a semantic tag (the greater the more confident the model is). According to [21], these coefficients, with initial values set to 1 for the first set of chunk/DA pairs extracted from the ontology, could be re-evaluated afterwards in light of users positive/negative feedbacks and estimated for newly introduced chunks during the online process. Then, the chunks in K are projected into F, as the sum of their composing words D-dimensional vector representations (embeddings).

Finally, the third component of the baseline ZSSP model is the parser itself. It composes a scored graph (finite-state machine) of semantic tag sequence hypotheses from any novel user utterance. All possible contiguous word sequences (chunks) are considered in the parsing algorithm (the combinatory explosion is limited in our case as the utterance length is generally below 10 words). For example if the user says "yeah downtown", as in Fig. 1, 3 different chunks are considered: "yeah", "downtown" and "yeah downtown". These chunks are mapped to the feature space F with the same method used to map K's chunks into F. The resulting real valued vectors (blue circles in Fig. 1) are then compared in terms of distance (e.g. cosine distance in our work) to the known chunk vectors (black crosses in Fig. 1). Then, a dot product between the k-most similar vectors and their corresponding assignment coefficients in K matrix is computed to attribute each chunk a list of ordered (and scored) semantic hypotheses. Then a best-path decoding on the finite-state machine (highlighted path in Fig. 1) derives the best semantic sequence hypothesis at the utterance level. Since we deal with edge weights closely related to distance, we employ a shortest-path strategy in this work.

3. ONLINE INTERACTIVE REFINEMENT PROBLEM

The main objective of the work presented in this paper is to extend the initial online adaptation strategy (described in [21]) to also allow concept creation and thus domain extension. In this preliminary study, we adopt a simple strategy based on an Adversarial Bandit algorithm to address the model refinement problem. Before going further in the formulation, we first introduce this problematic on a static case.

3.1. Static case

We posit that the system has the choice between several actions (aka the arms in the bandit literature) in order to improve the DAs automatically associated to a user utterance. So, we first need to define the considered action space. However, we can already foresee that each action implies the collaboration of the user at a different level. Therefore, we introduce a measure of user effort regarding the action chosen by the system, in the form of a cost function. We also define an inefficiency measure of the model that we should try to minimize over time. This metric allows us to quantify the model improvement made by a specific action. Finally the model Refinement problem is formulated as a linear optimization problem when the system has the full knowledge of the objective function.

Actions space and user effort cost function. Once the user provides the sentence, the system can choose one action (from a probability distribution) among a set $\mathcal{I}$ of M actions. In this preliminary setup, we consider a case where $M = 3$ and $\mathcal{I}$ can be defined as :

$$\mathcal{I} := \{\text{Skip, YesNoQuestions, AskAnnotation}\}.$$

Let $i \in \mathcal{I}$ be the action index. We assume that the user effort $\phi(i) \in \mathbb{N}$ can be measured by the number of exchanges between the system and the user in order to perform action i. The different actions and associated user efforts are described below:

- Skip: Skip the refinement process. The cost of this action is always set to 0 ($\phi(\text{skip}) = 0$).
- YesNoQuestions: Refine the model by considering yes/no user responses about the correctness of the detected DAs in the best semantic hypothesis. $\phi(\text{YesNoQuestions}) = 1$ for the whole sentence acceptance and in case of rejection $\phi(\text{YesNoQuestions})$ is equal to the number of confirmation requests ($+1$ per DA in the best semantic hypothesis).
- AskAnnotation: Ask the user to annotate the incoming utterance. $\phi(\text{AskAnnotation}) = 1$ if the sentence is accepted straightaway. Otherwise $\phi(\text{AskAnnotation}) \in \{2, 3, 4\}$ times the sum of annotated DAs (as given by a reference annotation in our simulated setup, see below). Here, we assume that the user first informs the system about the word boundaries of the concept he plans to annotate ($+1$), and then that the system sequentially asks for $acttype$, $slot$ and $value$ if necessary[1] ($+1$ per interim question). It should be noted that new $slot$s and $value$s can be added by the user in this way (domain extension).

Inefficiency measure. In order to estimate the model performance over the current user utterance without the need of semantic transcription, we choose to introduce a measure extracted dynamically from the parser output. To integrate a global minimisation optimisation problem (with cost function) inefficiency is measured instead of efficiency. So, let $d \in [0, 1]$ be the average weights of the edges[2] in the best path of the finite state machine used by the semantic parser (see Fig 1). As explained in Section 2, these weights correspond

[1] Some $acttype$s are empty, such as $hello()$ or just contain a unique slot without value such as $request(food)$

[2] By removing the word penalty applied in the semantic decoding

to the dot product of the cosine distance between each chunk considered in the best path and the assignment coefficients of the k most similar examples in K. Thus, the greater it is, the lower the model fits the utterance. Indeed, a high average weight traduces the fact there are no corresponding (close enough) known examples in K. Depending on the chosen refinement action i, d is updated to $d'(i) \in [0, 1]$ due to following model modification. Here we depict the main mechanisms:

- Skip: Since this action does not imply model changes, the inefficiency measure remains constant. Thus, $d'(Skip) = d$.
- YesNoQuestions: By using this action, each of the m DAs in the best semantic hypothesis will be confirmed or negated by the user. According to [21], these user feedbacks are converted into a set U of m tuples $U := ((c_l, T_l, f_l))_{1 \leq l \leq m}$, where ($c_l$, T_l) is a chunk/tag pair proposed to the user and f_l is her feedback (1 positive, 0 negative). Given K and U after each interaction, the algorithm presented in [21] is used to update K into K'. Basically, these tuples are used to update the positive and negative counts observed up to now for the involved chunks[3] and exploit vicinity in the semantic space F to fill the unknown assignation values. So, $d'(YesNoQuestions) = \delta$ where δ is the new average weight of the utterance with newly updated K^*.
- AskAnnotation: If the sentence is accepted straightaway we consider that all the chunk/tag pairs extracted from the best semantic hypothesis received a positive user feedbacks. Otherwise, we employ the m' annotated chunk/tag pairs[4] are cast into tuples $U := ((c_l, T_l, 1)_{1 \leq l \leq m'})$. In this specific case, new semantic output tags and values can be added. Due to the fact that parts of the user utterance are now in K' as positive examples, $d'(AskAnnotation) \approx 0$.

Loss function. Finally we need to define a loss function such that the system, by optimize it, will minimize at the same time updated inefficiency measure $d'(i)$ and the user effort $\phi(i)$. Thus we propose to define the loss function $l(i) \in [0, 1]$ as the convex combination of the two measures:

$$l(i) := \underbrace{\gamma d'(i)}_{\text{system improvement}} + \underbrace{(1-\gamma)\frac{\phi(i)}{\phi_{max}}}_{\text{user effort}},$$

where $\gamma \in [0, 1]$ balance the importance of information improvement and user effort for the system and $\phi_{max} \in \mathbb{N}_+$ is the maximum number of exchanges between the system and the user (in a same turn/round). Let $\mathbf{p} \in \Delta(3) := \{\mathbf{q} \in \mathbb{R}^3_+ \mid \sum_{i \in \mathcal{I}} q(i) = 1\}$ the probability distribution over the different actions. The Refinement objective is defined as:

$$\min_{\mathbf{p} \in \Delta(3)} E[l] = \sum_i p(i) l(i).$$

If we have a full knowledge of the cost $l(i)$ for each action i, the Refinement problem is equivalent to solve $\min_i\{l(i)\}$. However in a real scenario, this framework cannot be applied as the loss function $l(i)$ is not known. For instance, when the system uses $i = YesNoQuestions$, $d'(YesNoQuestions)$ and $\phi(YesNoQuestions)$ will be revealed only after the action is made. Moreover the system receives several sentences, at different times, from different users. Thus comes the need to adapt the Refinement model to an adaptive framework (Adversarial bandit scenario).

[3] Initial set of chunk/tag pairs is initialized by considering these examples as confident and positive user feedbacks

[4] Notice that m' may not be equal to m because the user is able to change the DA boundaries

3.2. Adversarial Bandit case

In this paper we cast the AL problem of the SLU module into an adversarial bandit problem. We consider the following scenario:

The adversarial bandit refinement problem

Known parameters: Set of actions $\mathcal{I}$ and parameter $\gamma \in [0, 1]$. At each turn/round $t = 1, 2, \ldots$

1. The system receives a user utterance and computes d_t;
2. The system chooses an action $i_t \in \mathcal{I}$, possibly with the help of external randomization;
3. Once the action i_t is performed, the system computes:
   - The inefficiency measure $d'_t(i_t)$ with the collaboration of the user;
   - The user effort $\phi_t(i_t)$, which is the exchange count between the system and the user to compute i_t;
   - The current loss is finally

$$l_t(i_t) = \gamma d'_t(i_t) + (1-\gamma)\phi_t(i_t).$$

Goal: Find $i_1, i_2, \ldots$, such that for each T, the system minimizes the total loss:

$$\sum_{t=1}^{T} l_t(i_t) = \gamma \sum_{t=1}^{T} d'_t(i_t) + (1-\gamma) \sum_{t=1}^{T} \phi_t(i_t).$$

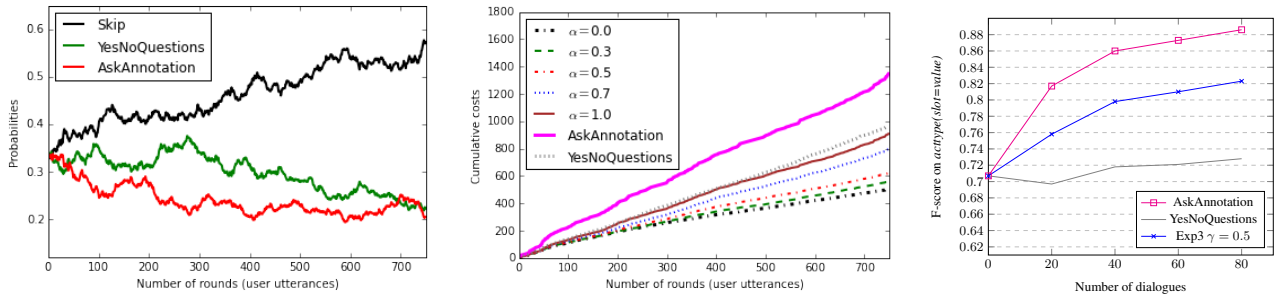
No assumptions are made about $d'_t(i_t) \in [0, 1]$ and $\phi_t(i_t) \in [0, 1]$. Thus we do not consider that the action i_{t-l}, with $l \in \{1, \ldots, t-1\}$, has a specific effect over the current loss function at round t. This hypothesis is coming from the fact that the user sentence cannot be accurately predicted without some strong priors. This scenario is thus closely related to a cold start problem. These previous remarks lead us to choose the Adversarial Bandit framework instead of the stochastic one.

An efficient algorithm to solve the Adversarial Bandit problem with a small number of arms is the Exp3 [24]. A mathematical proof of the relative high performance of this algorithm, in term of regret optimisation, has been proposed in the monograph [25].

3.3. Simulated environment

In order to thoroughly test the policy learning algorithm we choose in this preliminary study to simulate the user responses. To do so, we have implemented an annotation evolution indicator able to determine the correctness of the machine proposition according to a given reference. Due to the fact that $acttype(slot = value)$ semantic tags are not aligned with words in the considered corpus and since a word level tagging is a prerequisite to annotate with chunk/tag pairs, we choose to use an adapted unsupervised alignment procedure following [29]. Thus, at each turn we have sufficient information to be able to respond accurately to the machine action (reference DAs and their alignments with words).

The simulated user employs 3 different actions. *Affirm* and *Negate* are used to respond to a confirmation machine action



(a) (b) (c)

Fig. 2: **(a)** Exp3 probability distribution over the action set **(b)** Impact of γ over the cumulative user efforts (cumulative costs) **(c)** Impact of the number of dialogues handled by the online adaptive strategies on the F-score metric (DSTC2 test set).

(for AskAnnotation and YesNoQuestions). *Inform* are used exclusively for AskAnnotation (e.g. *Inform(acttype=request)*, *Inform(boundaries="austrian food")*). Here, we assume that annotation subdialogues could be handle by the system which a high level of accuracy (e.g. by using a well calibrated grammar and a finely-tuned strategy).

4. EXPERIMENTS AND RESULTS

Experiments description. All experiments are based on the DSTC2 datasets [27] covering the domain of restaurant search. The research challenge focused on tracking the user's goal all along the dialogue, here we only consider the SLU subtask. Thus, we exploit the fully annotated data (e.g. transcriptions, dialogue-act semantics) as training and testing sets to evaluate our online learning approach on realistic dialogue settings. The approach is evaluated on the given transcriptions of the challenge test set (9890 user utterances). A subset of transcriptions from the DSTC2 training set (1472 transcribed user utterances) is exploited to evaluate the online refinement model.

The ZSSP model used in the online learning is initialised as in [21]. A 300-dimensional word2vec [23] word-embeddings model is trained on a large amount of wide coverage and freely available English corpora with the Skip-gram algorithm (10-word window). The resulting model is expected to exhibit some linguistic regularities [30] as well as a linear structure. Hence it is possible to combine the words by an element-wise addition of their word embeddings [31]. The technique is employed to directly map examples in K to their corresponding word2vec representation seen as the sum of individual word representations. As the cosine similarity/distance has been proved to be well adapted to the word2vec model [23, 30], this metric is used in a k-nearest neighbours classifier for the chunk prediction and update. In the following experiments $k = 1$ and word penalty is set to 0.5 for parsing and 20-nearest neighbours strategy is employed to estimate unobserved assignment coefficients.

The task-dependent knowledge base used in these experiments is derived from the DSTC2 challenge ontology and from few generic dialogue knowledge. The semantics of the DSTC2 task is represented by 16 different act types, 8 slots and 215 values. In the considered ontology, slots and values have already lexicalised names (e.g. "address", "french", etc.). The lexical forms (53) used to model task-independent act types were manually written (for example "say again" for the `repeat` act). In this work we deliberately degraded K by removing important slots such as *name* and *signature* and values Thus we start with a relatively low F-score of 0.70 on the DSTC2 test transcription. Overall, 404 chunks are considered and assigned to 78 (out of 663 possible) different semantic tags. Since Exp3 employs some form of stochastic exploration, we simulate 20 independent online learning processes in parallel and average their figures in all the results presented hereafter.

Experiments results. Figure 2(a) gives the evolution of the probability $\mathbf{p}_t$ associated to each action as provided by Exp3. We can observe that first each action is selected with comparable probability, Exp3 is exploring. Then, as the number of turns increases, we observe the rising influence of both YesNoQuestions and Skip actions which finally ends up with a clear advantage to the Skip action when new information become harder to collect with respect to the cost involved.

In Figure 2(b) we compare the effect of γ on the Exp3 exploration strategy, in terms of cumulative user effort, with the AskAnnotation and YesNoQuestions baselines (i.e. always performing these actions). The measure is depicted for $\gamma \in \{0, 0.3, 0.5, 0.7, 1\}$. We can observe that AskAnnotation is the more expensive followed by YesNoQuestions. Varying the γ tends to have a positive effect on the evolution of the learning action. The greater γ is, the less costly the learning action is. When the cost is ignored in the loss function ($\gamma = 1.0$), Exp3 tends to use costly action that reduce the distance but it still can learn that YesNoQuestions are as useful as AskAnnotation at some point. Thus, γ could allow to tune some application specific trade-off between user effort and model efficiency.

Finally, in Figure 2(c) Exp3 is compared in terms of F-score on DSTC2 test set against the AskAnnotation and YesNoQuestions baselines. As expected AskAnnotation performs the best. Indeed, the use of new annotations allows ZSSP to dynamically cover additional DAs and update K with robust examples. Due to the fact that the goal of Exp3 is to find a trade-off between lower user effort and model efficiency, this method is able to reach in a cost effective manner performances closer to those obtained with AskAnnotation compared with YesNoQuestions.

5. CONCLUSION

In this paper an Adversarial Bandit approach to refine a zero-shot learning SLU, ZSSP, is proposed and tested in order to alleviate a limited coverage on the domain specific semantics. It has been proved to be efficient and to provide a practical way to formalise a trade-off between user supervision effort and system efficiency. Comparisons with other bandit techniques as well as generalisation of the approach (e.g. extended action set) are in progress. Likewise integration in a live dialogue system with seed expert users is an ongoing work to study the effect over the overall dialogue progress (task completion and user satisfaction) and the relation with the dialogue manager strategy learning.

6. REFERENCES

[1] S. Hahn, M. Dinarelli, C. Raymond, F. Lefèvre, P. Lehnen, R. De Mori, A. Moschitti, H. Ney, and G. Riccardi, "Comparing stochastic approaches to spoken language understanding in multiple languages," *IEEE TASLP*, vol. 19, no. 6, pp. 1569–1583, 2010.

[2] Y. He and S. Young, "Semantic processing using the hidden vector state model," *Computer Speech and Language*, vol. 19, pp. 85–106, 2005.

[3] F. Lefèvre, "Dynamic Bayesian networks and discriminative classifiers for multi-stage semantic interpretation," in *ICASSP*, 2007.

[4] A. Deoras and R. Sarikaya, "Deep belief network based semantic taggers for spoken language understanding," in *INTERSPEECH*, 2013.

[5] N. Camelin, B. Detienne, S. Huet, D. Quadri, and F. Lefèvre, "Unsupervised concept annotation using latent dirichlet allocation and segmental methods," in *EMNLP Workshop on Unsupervised Learning in NLP*, 2011.

[6] Y. Gao, L. Gu, and H.K.J. Kuo, "Portability challenges in developing interactive dialogue systems," in *ICASSP*, 2005.

[7] A. Celikyilmaz, G. Tur, and D. Hakkani-Tur, "Leveraging web query logs to learn user intent via bayesian latent variable model," in *ICML*, 2011.

[8] D. Hakkani-Tur, L. Heck, and G. Tur, "Exploiting query click logs for utterance domain detection in spoken language understanding," in *ICASSP*, 2011.

[9] L. Heck and D. Hakkani-Tur, "Exploiting the semantic web for unsupervised spoken language understanding," in *SLT*, 2012.

[10] G. Tur, D. Hakkani-tur, D. Hillard, and A. Celikyilmaz, "Towards unsupervised spoken language understanding: Exploiting query click logs for slot filling," in *INTERSPEECH*, 2011.

[11] A. Lorenzo, L. Rojas-Barahona, and C. Cerisara, "Unsupervised structured semantic inference for spoken dialog reservation tasks," in *SIGDIAL*, 2013.

[12] F. Lefèvre, D. Mostefa, L. Besacier, Y. Esteve, M. Quignard, N. Camelin, B. Favre, B. Jabaian, and L. Rojas-Barahona, "Robustness and portability of spoken language understanding systems among languages and domains: the PORT-MEDIA project," in *LREC*, 2012.

[13] F. Lefèvre, F. Mairesse, and S. Young, "Cross-lingual spoken language understanding from unaligned data using discriminative classification models and machine translation.," in *INTERSPEECH*, 2010.

[14] B. Jabaian, L. Besacier, and F. Lefèvre, "Comparison and Combination of Lightly Supervised Approaches for Language Portability of a Spoken Language Understanding System," *IEEE TASLP*, vol. 21, no. 3, pp. 636–648, 2013.

[15] G. Tur, G. Rahim, and D. Hakkani-Tur, "Active labeling for spoken language understanding.," in *EUROSPEECH*, 2003.

[16] P. Gotab, F. Béchet, and G. Damnati, "Active learning for rule-based and corpus-based spoken language understanding models," in *ASRU*, 2009.

[17] F García, L. Hurtado, E. Sanchis, and E. Segarra, "An active learning approach for statistical spoken language understanding," in *CIARP*, 2011.

[18] R. Sarikaya, "Rapid bootstrapping of statistical spoken dialogue systems," *Speech Communication*, vol. 50, no. 7, pp. 580–593, 2008.

[19] A. Bayer and G. Riccardi, "On-line adaptation of semantic models for spoken language understanding," in *ASRU*, 2013.

[20] P. Gotab, G. Damnati, F. Béchet, and L. Delphin-Poulat, "Online slu model adaptation with a partial oracle," in *INTERSPEECH*, 2010.

[21] E Ferreira, B Jabaian, and F Lefèvre, "Online adaptative zero-shot learning spoken language understanding using word-embedding," in *ICASSP*, 2015.

[22] E Ferreira, B Jabaian, and F Lefèvre, "Zero-shot semantic parser for spoken language understanding," in *INTERSPEECH*, 2015.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[24] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, pp. 48–77, 2003.

[25] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[26] L. Ralaivola, B. Favre, P. Gotab, F. Béchet, and G. Damnati, "Applying multiclass bandit algorithms to call-type classification," in *Workshop on Automatic Speech Recognition & Understanding, ASRU*, 2011.

[27] M. Henderson, B. Thomson, and J. Williams, "The second dialog state tracking challenge," in *SIGDIAL*, 2014.

[28] J. Bian, B. Gao, and T. Liu, "Knowledge-powered deep learning for word embedding," in *ECML*, 2014.

[29] Stéphane Huet and Fabrice Lefèvre, "Unsupervised alignment for segmental-based language understanding," in *Proceedings of the First Workshop on Unsupervised Learning in NLP*, 2011.

[30] T. Mikolov, W. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *NAACL-HLT*, 2013.

[31] T. Mikolov, I. Sutskever, K. Chen, G. S Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013.