Simulating ASR errors for training SLU systems

Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, Yannick Estève
LIUM, Le Mans University, France
firstname.lastname@univ-lemans.fr

Abstract

This paper presents an approach to simulate automatic speech recognition (ASR) errors from manual transcriptions and describes how it can be used to improve the performance of spoken language understanding (SLU) systems. In particular, we point out that this noising process is very usefull to obtain a more robust SLU system to ASR errors in case of insufficient training data or more if ASR transcriptions are not available during the training of the SLU model. The proposed method is based on the use of both acoustic and linguistic word embeddings in order to define a similarity measure between words dedicated to predict ASR confusions. Actually, we assume that words acoustically and linguistically close are the ones confused by an ASR system. By using this similarity measure in order to randomly substitute correct words by potentially confusing words in manual annotations used to train CRF- or neural- based SLU systems, we augment the training corpus with these new noisy data. Experiments were carried on the French MEDIA corpus focusing on hotel reservation. They show that this approach significantly improves SLU system performance with a relative reduction of 21.2% of concept/value error rate (CVER), particularly when the SLU system is based on a neural approach (reduction of 22.4% of CVER). A comparison to a naive noising approach shows that the proposed noising approach is particularly relevant.

Keywords: spoken language understanding, data augmentation, noising, automatic speech recognition, errors

1. Introduction

Spoken language understanding (SLU) consists in extracting semantic information from speech, and can refer to different tasks. In (De Mori, 2007), the author defines it as "(...) the interpretation of signs conveyed by a speech signal". Similar to previous works from other authors (Hahn et al., 2011; Mesnil et al., 2015), the SLU task targeted in this paper consists in automatically extracting semantic concepts and concept/values pairs from the automatic transcriptions in order to feed a dialogue manager. This task can also be perceived as a slot filling task.

Usually, SLU needs first an automatic transcription of user utterances thanks to an automatic speech recognition (ASR) system. These recognized words are then analyzed in order to extract their meanings. Even best SLU systems see their performance drop when making the transition from processing manual transcriptions to automatic ones, since ASR errors make the SLU task harder. In order to reduce this unavoidable performance decline, it would be relevant to prepare SLU systems to ASR errors during their training. For instance, it is known that Spoken Dialog System (SDS) applied to automatic transcriptions perform better when they are trained on automatic transcriptions rather than manual ones. Since large automatic transcription corpora needed for SDS training are rare, some approaches have been presented in order to simulate ASR errors to train these SDS (Pietquin and Beaufort, 2005; Schatzmann et al., 2007). Such ASR error simulation has also been applied to train discriminative language models in order to improve ASR performance in terms of word error rate (Jyothi and Fosler-Lussier, 2010).

Nowadays, SLU systems are often built through a data-driven approach (Raymond et al., 2006; De Mori et al., 2008; Hahn et al., 2011; Sarikaya et al., 2014; Mesnil et al., 2015; Hakkani-Tür et al., 2016). For slot/filling tasks, manual annotations are usually produced to tag manual transcriptions with semantic labels in order to build a training corpus. In the study presented in this paper we make the assumption – and check it – that building SLU systems from automatic transcriptions yields to SLU systems more robust to ASR errors. Nevertheless, getting automatic transcriptions implies the availability of audio recordings related to the manual semantic annotations, and the availability of an ASR system. More, to get an effective ASR system, some training or adaptation data are needed to tune it while these data are usually the same as the ones used to train the SLU module: this implies to manipulate these data very carefully, in order to avoid biases coming, for instance, from overfitting.

Our objective is to propose an approach to simulate ASR errors from manual transcriptions, in order to create a SLU training corpus closer to the data that the SLU system will have to process on test. In that way, robust SLU systems can be trained even if no ASR data on the specific task is available. This simulation consists in introducing errors in a manual corpus by substituting correct words by similar ones. We assume that words confusable by an ASR system are words that are acoustically close. Such assumption was also retained in (Fosler-Lussier et al., 2002; Stuttle et al., 2004), where ASR simulation was based on the similarity of the phonetization of words to evaluate their confusability. We also consider that these confusable words are also linguistically close. To compute a confusability measure between words, we present in this paper a new approach based on the use of both acoustic and linguistic word embeddings. In our experiments, we measure the impact of this ASR simulation used to modify the training corpus of two different data-driven SLU architectures: one based on conditional random fields (Lafferty et al., 2001) (CRF) and the other one based on a neural network encoder-decoder with attention mechanism (Cho et al., 2014) (NN-EDA). These experiments are carried on the French MEDIA cor-

pus, on which CRF still perform better than neural approaches (Vukotic et al., 2015; Simonnet et al., 2017).

2. ASR confusability measure and simulation of ASR errors

The proposed confusability measure is based on the use of linguistic and acoustic similarities. These similarities are computed from cosine similarities between linguistic and acoustic word embeddings.

The linguistic word embeddings correspond to a combination through a principal component analysis (PCA) of different kinds of word embeddings: *word2vecf* on dependency trees (Levy and Goldberg, 2014), *skip-gram* provided by *word2vec* (Mikolov et al., 2013), and *GloVe* (Pennington et al., 2014), as described in (Ghannay et al., 2016).

The acoustic embeddings correspond to the projection of an arbitrary or fixed dimensional speech segment in a fixed-dimensional space, in a manner that preserves acoustic similarity between words. The approach used to build the acoustic word embeddings was proposed by (Bengio and Heigold, 2014).

2.1. Linear interpolation of linguistic and acoustic similarities

In this study, we propose to use linguistic and acoustic word embeddings to predict ASR confusions. With the purpose to take benefit from both linguistic and acoustic similarities, we propose to use a linear interpolation to combine them. This results to a similarity called $LA_{SimInter}$, defined as:

$$LA_{SimInter}(\lambda, x, y) = (1-\lambda)\times L_{Sim}(x, y)+\lambda\times A_{Sim}(x, y)$$

where x and y are two words, λ is the interpolation coefficient, while L_{Sim} and A_{Sim} are respectively the linguistic and acoustic similarities computed with the cosine similarity applied to respectively the linguistic and acoustic word embeddings of x and y.

Since our goal is to predict ASR confusions, we aim to optimize the λ value for this purpose. To estimate λ, a list of known substitution errors made by an ASR system is used. Let define h an erroneous word hypothesis and $\overline{r}$ the reference word that is substituted with h.

For each word pairs $(h, \overline{r})$ in the list, we compute the probability of using h when the reference word $\overline{r}$ is wrong, *i.e.* the probability of substituting the reference word with the hypothesis one, which is defined as:

$$P(h|\overline{r}) = \frac{\#(h, \overline{r})}{\#\overline{r}}$$

where $\#(h, \overline{r})$ refers to the number of occurrences of the word pair and $\#\overline{r}$ is the number of errors (deletion + substitution) on the reference word.

Based on the similarity score $LA_{SimInter}(h, \overline{r})$ and the probability $P(h|\overline{r})$, we choose the interpolation coefficient $\hat{\lambda}$ that minimizes the mean squared error (MSE) such as:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}}\, MSE(\forall(h, \overline{r}) : P(h|\overline{r}), LA_{SimInter}(\lambda, h, \overline{r}))$$

By using $LA_{SimInter}$ with $\hat{\lambda}$, it is now possible to propose for a given word its linguistically and acoustically nearest neighbors. We consider the value of $LA_{SimInter}(\hat{\lambda}, x, y)$ as a confusability measure between words x and y, and we call it $confus(x, y)$.

2.2. Simulating errors

To simulate ASR errors, we apply the confusability measure $confus(x, y)$ in order to substitute some correct words from manual transcriptions by one of its confusable words. By fixing a targeted word error rate e (only substitutions), we randomly modify e percent of occurrences of words. These substitutions are made after defining two thresholds: the value c that refers to the lowest value of $confus(\overline{r}, h)$ that permits to substitute the word $\overline{r}$ by the confusable word h, and the value n that limits the number of the possible substitutions of $\overline{r}$ to the n closest h_i words (*i.e.* the words h_i such as the $confus(\overline{r}, h_i)$ value is one of the n highest values for a given $\overline{r}$). The word h is randomly chosen from the list of h_i words that respect the constraints of the n and c thresholds

3. Experimental Setup

This section describes the experimental setup of our work, which is inspired from our previous study (Simonnet et al., 2017).

3.1. The MEDIA corpus

The corpus used here is the MEDIA corpus, collected in the French Media/Evalda project (Bonneau-Maynard et al., 2005) and dealing with negotiation of tourist services. It contains three sets of telephone human/computer dialogues, namely: a training set (TRAIN) with approximately 17.7k sentences, a development set (DEV) with 1.3k sentences and an evaluation set (TEST) containing 3.5k sentences. The corpus was manually annotated with semantic concepts characterized by a label and its value. Evaluations are performed with the DEV and TEST sets and report concept error rates (CER) for concept labels only and concept-value error rates (CVER) for concept-value pairs. It is worth mentioning that the number of concepts annotated in a turn has a large variability and may include more than 30 annotated concepts.

For these experiments, a variant of the ASR system developed by LIUM that won the last evaluation campaign on French language has been used (Rousseau et al., 2014). This system is based on the Kaldi speech recognition toolkit (Povey et al., 2011). A detailed description of the ASR system is given in our previous study (Simonnet et al., 2017).

The word error rates for the training, development, and test corpora are respectively 23.7%, 23.4% and 23.6%.

3.2. SLU features and architectures

Two basis SLU architectures are considered to carry experiments on the MEDIA corpus. The first one is an encoder/decoder recurrent neural architecture with a mechanism of attention (NN-EDA) similar to the one used for machine translation proposed in (Cho et al., 2014). The second one is based on CRF. Both architectures build their training model on the same features encoded with continuous values in the first one and discrete values in the second one.

3.2.1. Set of Features

Word features are added in input with the words. They help the SLU system to achieve better understanding, in-

spired from (Hahn et al., 2011). The features used here (for a given word) are the following: its pre-defined semantic categories which are the MEDIA specific categories and more general categories; sets of syntactic and morphological features; and two ASR confidence measures. The confidence measures are the ASR *posterior* probability (*pap*) and the Multi-Stream Multi-Layer Perceptron (MS-MLP) confidence measure as described in our previous work (Simonnet et al., 2017). Both of these features are an estimation of the reliability of the recognized word.
The detailed description of these features is described in (Simonnet et al., 2017).
The two SLU architectures take all these features except for the confidence measures where only one is taken for a purpose of experimental consistency as it will be described in subsection 3.3.. These architectures also need to be calibrated on their respective hyper-parameters in order to give the best results. The way the best configuration is chosen is described in 4..

3.2.2. Neural Network EDA system

The proposed NN-EDA system, which is inspired from a machine translation architecture, was implemented by starting from the *nmtpy* toolkit (Caglayan et al., 2017). The concept tagging process is considered as a translation problem from words (source language) to semantic concept tags (target language).
The bidirectional RNN encoder is based on Gated Recurrent Units (GRU) and computes annotations for each word from the input sequence. These annotations are the concatenation of the matching forward and backward hidden layer states obtained respectively by the forward and the backward RNN comprising the bidirectional RNN. Thus they contain the summaries of the dialogue turn contexts respectively preceding and following a considered word.
The sequence of annotations is then used by the decoder to compute a context vector (recomputed after each emission of an output label). This computation takes into account a weighted sum of all the annotations computed by the encoder. This weighting depends on the current output target, and is the core of the attention mechanism: a good estimation of these weights allows the decoder to choose parts of the input sequence to pay attention to, in order to make a decision about the current label output.
A more detailed description of the NN-EDA system is given in (Simonnet et al., 2017).

3.2.3. CRF system

Past experiments described in (Hahn et al., 2011) have shown that the best semantic annotation performance on manual and automatic transcriptions of the MEDIA corpus were obtained with CRF systems. More recently in (Vukotic et al., 2015), this architecture has been compared to popular bi-directional RNN (bi-RNN). The result was that CRF systems outperform a bi-RNN architecture on the MEDIA corpus, while better results were observed by bi-RNN on the ATIS (Hemphill et al., 1990) corpus. This is probably explained by the fact that MEDIA contains semantic contents whose mentions are more difficult to disambiguate, and CRFs make it possible to exploit complex contexts more efficiently.
For the sake of comparison with the best SLU system proposed in (Hahn et al., 2011), the Wapiti toolkit was used (Lavergne et al., 2010) in our study. Nevertheless, the set of input features used by the system proposed in this paper is different from the one used in (Hahn et al., 2011). Among the novelties used in our system, we consider syntactic and ASR confidence features and our configuration template is different. After many experiments performed on DEV, our final feature template includes the previous and following instances for words and POS in a unigram or a bigram to associate a semantic label with the current word. Also associated with the current word are semantic categories of the two previous and two following instances. The other features are only considered at the current position.
Furthermore, the tool discretize4CRF[1] is used to apply a discretization function to the ASR confidence measures in order to obtain several discrete values that can be accepted as input features by the CRF.

3.3. ASR simulation

We applied the method presented in sub-section 2.2. in order to simulate ASR errors. Starting from the manual annotations of the MEDIA corpus (without error), we build different datasets. In these simulations we fixed the e value to 20, that represents the rate of words we corrupt randomly in the manual transcriptions.
Two different simulations were tested, by choosing different threshold values n and c;

- **N.7 corpus**: $n = 7$ and $c = 0.4$;
- **N.10 corpus**: $n = 10$ and $c = 0.5$.

Another artificial dataset was created, called **noise.naive corpus**: this corpus does not take into account of the confusability measure. In this dataset, the same $e = 20$ percent of words from manual transcriptions are randomly substituted, by simply choosing randomly a word from the entire MEDIA vocabulary. When a correct word is substituted with a confusable one, we use their confusability measure as an ASR confidence measure. For a purpose of experimental consistency, when working on ASR, we only give one ASR confidence measure among the two available ones in order to always have the same number of confidence measure.

4. Experimental results

Experiments were carried with the MEDIA corpus. Both SLU architectures are optimized to get the best CVER. The training is done on the TRAIN set. For the NN-EDA, validations during training are performed on the DEV set in order to choose the best parameters.
Results on TEST in terms of CER and CVER are reported in tables 1 and 2, where **M** refers to manual corpus, **A** to a corpus composed by automatic transcriptions, and **N** to a noised corpus. TEST corpus is made by ASR transcriptions only, while the nature of TRAIN or DEV corpora varies in our experiments.

[1] https://gforge.inria.fr/projects/discretize4crf/

4.1. Tuning on ASR transcriptions

Since evaluation on TEST is made on ASR transcriptions, we first consider that a DEV corpus composed of automatic transcriptions is available to tune our SLU systems. Such corpus is less hard to collect than a training corpus, since it contains only about 1300 sentences (in comparison to the 17700 in the MEDIA training corpus). Above all, as evocated in the introduction, processing data that are outside the training corpus is easier since no problem of bias and overfitting can be introduced.

Experimental results in this configuration are visible in table 1. We can first notice that our assumption on the importance of getting automatic or ASR simulated transcriptions to get training data as close as possible to the test data is checked: with an **A** TRAIN set, results for both NN-EDA and CRF are very significantly better than with the use of a **M** TRAIN set. It also appears that the CRF architecture significantly outperforms NN EDA for both **M** and **A** training corpora. It is also clear that training a SLU system on manual transcriptions only is largely insufficient to handle ASR transcriptions. The system needs to be prepared to ASR errors.

	NN-EDA		*CRF*	
TRAIN set	*CER*	*CVER*	*CER*	*CVER*
M	31.6	36.2	27.5	31.6
A	22.5	28.3	19.9	25.1
N.7	23.8	29	22.6	27.7
Double N.7	23.2	28.8	26.3	31.3
M+N.7	22.7	28.1	22.6	27.7
M+N.10	23.3	28.5	23.2	28.3
M+N.naive	23.7	28.8	25	30.3
M+A	20.7	25.8	20.2	25.3
M+N.7+A	20.2	26	29.1	33.0

Table 1: Comparison on CER and CVER obtained on ASR TEST with an ASR DEV.

Training however on a noised corpus (line N.7) gets interesting results. It clearly gets an improvement from the poor results gotten on manual transcriptions only. It gets close to the results of using pure ASR transcription and so confirm that our approach to simulate ASR errors seems acceptable for this task. Training on a double noised corpus (line Double N.7, in which two successive ASR error simulations on the same train were applied) can improve a little the results on the NN-EDA while it decreases strongly the CRF results.

Better results can be achieved by combining manual and noised corpus. By using the N.7 dataset in combination with the manual corpus, the results are just as good as the pure ASR for the NN-EDA. The CRF gets the same results as for N.7 only.

We can also see the comparison between the different types of noise. The N.7 gets better results than N.10 showing that by substituting correct words with globally less similar words decreases the results. Furthermore, even if applying naive noise gets better results than using manual transcriptions without errors, we obtain the worst scores among noising approaches with it. This globally shows the importance of an intelligently generated noise, and implicitly validates our ASR error simulation approach.

Finally the best achieved results that outperform even pure ASR alone are obtained by training the SLU system on a combination of ASR and manual corpus. Both SLU systems find their best results in this configuration and the gap between CRF and NN-EDA has been strongly reduced from the experiments on ASR only or manual only. Training on a triple combination of manual, ASR and noised corpus does not increase more these results.

In general, CRF significantly outperforms NN-EDA when these systems are trained from a manual or an ASR corpus. But NN-EDA takes better benefit from ASR simulation, or from manual and ASR combination than CRF. At the end, best results for both NN-EDA and CRF are now very close, showing some potentialities of neural networks, not shared by CRF, to learn relevant information from noisy data.

4.2. Tuning on manual transcriptions

In this section we explore the scenario in which ASR data are *not* available to tune the SLU system. In that case, this explicitly means that the DEV corpus can not be issued from ASR. This can become problematic when the SLU system needs to compute some validations during training, which is the case for NN-EDA. CRF otherwise do not use the DEV while training. Thus the results visible in table 2 are only for NN-EDA (the CRF scores stay unchanged).

		NN EDA	
TRAIN set	*DEV set*	*CER*	*CVER*
M	M	33.9	38.2
N.7	N.7	23.5	28.6
M+N.7	N.7	23.1	28.5

Table 2: Comparison on CER and CVER obtained on ASR TEST with no ASR TRAIN or DEV.

In general, except for the noised train corpus alone, it gives better results to validate on an ASR DEV corpus, closer in nature to the TEST data.

Nevertheless, even if those results are a little bit worse than the ones reached by validating on an ASR DEV corpus, we can notice that it is possible to very significantly improve the SLU systems by applying our ASR error simulation approach in order to enrich or to noise the SLU training and development data which are composed by only manual transcriptions.

5. Conclusion

Two SLU architectures based on NN-EDA and CRF were compared in this study. An ASR error simulation based on a confusability measure built from acoustic and linguistic word embeddings has been proposed and used in order to noise a manual annotated corpus. Experiments show that this noising process is relevant to enrich and to prepare an SLU training corpus. If no ASR system is available to prepare these data, our proposition offers a very significant improvement of the SLU performances, from 36.2% of CVER with only manual annotations in the training corpus, in

comparison to 28.5% of CVER by applying our approach: this represents a relative reduction of 21.2% of the concept/value errors. Another interesting result of this study is the contraction of the differences, in terms of CER or CVER, between CRF and NN-EVA on the MEDIA corpus. No advance on this corpus has been made from 2011 (Hahn et al., 2011) and CRF are still dominant. Our results show that it is now possible to get similar results with neural network architecture. We expect to propose new contributions to make neural networks more effective than CRF, that have reached a plateau several years ago on this task. In a close future, we will also consider other ASR error simulation approaches to compare their impact to ours to prepare and enrich SLU training corpus. We will also experiment the use of our ASR simulation on other tasks, like ASR error detection for instance.

6. Bibliographical References

Bengio, S. and Heigold, G. (2014). Word embeddings for speech recognition. In *INTERSPEECH*, pages 1053–1057.

Bonneau-Maynard, H., Rosset, S., Ayache, C., Kuhn, A., and Mostefa, D. (2005). Semantic annotation of the french media dialog corpus. In *Ninth European Conference on Speech Communication and Technology*.

Caglayan, O., García-Martínez, M., Bardet, A., Aransa, W., Bougares, F., and Barrault, L. (2017). Nmtpy: A flexible toolkit for advanced neural machine translation systems. *arXiv preprint arXiv:1706.00457*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

De Mori, R., Bechet, F., Hakkani-Tur, D., McTear, M., Riccardi, G., and Tur, G. (2008). Spoken language understanding. *IEEE Signal Processing Magazine*, 25(3).

De Mori, R. (2007). Spoken language understanding: A survey. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 365–376. IEEE.

Fosler-Lussier, E., Amdal, I., and Kuo, H.-K. J. (2002). On the road to improved lexical confusability metrics. In *ISCA Tutorial and Research Workshop (ITRW) on Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology*.

Ghannay, S., Favre, B., Esteve, Y., and Camelin, N. (2016). Word embedding evaluation and combination. In *of the Language Resources and Evaluation Conference (LREC 2016), Portoroz (Slovenia)*, pages 23–28.

Hahn, S., Dinarelli, M., Raymond, C., Lefevre, F., Lehnen, P., De Mori, R., Moschitti, A., Ney, H., and Riccardi, G. (2011). Comparing stochastic approaches to spoken language understanding in multiple languages. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1569–1583.

Hakkani-Tür, D., Tur, G., Celikyilmaz, A., Chen, Y.-N., Gao, J., Deng, L., and Wang, Y.-Y. (2016). Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*.

Hemphill, C. T., Godfrey, J. J., Doddington, G. R., et al. (1990). The atis spoken language systems pilot corpus. In *Proceedings of the DARPA speech and natural language workshop*, pages 96–101.

Jyothi, P. and Fosler-Lussier, E. (2010). Discriminative language modeling using simulated asr errors. In *Eleventh Annual Conference of the International Speech Communication Association*.

Lafferty, J., McCallum, A., Pereira, F., et al. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July.

Levy, O. and Goldberg, Y. (2014). Dependency based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 302–308.

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., et al. (2015). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):530–539.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, volume 12.

Pietquin, O. and Beaufort, R. (2005). Comparing asr modeling methods for spoken dialogue simulation and optimal strategy learning. In *Ninth European Conference on Speech Communication and Technology*.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number EPFL-CONF-192584. IEEE Signal Processing Society.

Raymond, C., Bechet, F., De Mori, R., and Damnati, G. (2006). On the use of finite state transducers for semantic interpretation. *Speech Communication*, 48(3):288–304.

Rousseau, A., Boulianne, G., Deléglise, P., Estève, Y., Gupta, V., and Meignier, S. (2014). LIUM and CRIM ASR system combination for the REPERE evaluation campaign. In *International Conference on Text, Speech, and Dialogue*, pages 441–448. Springer.

Sarikaya, R., Hinton, G. E., and Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(4):778–784.

Schatzmann, J., Thomson, B., and Young, S. (2007). Error

simulation for training statistical dialogue systems. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 526–531. IEEE.

Simonnet, E., Ghannay, S., Camelin, N., Esteve, Y., and Renato, D. M. (2017). ASR error management for improving spoken language understanding. In *INTERSPEECH*.

Stuttle, M., Williams, J., and Young, S. (2004). A framework for dialog systems data collection using a simulated asr channel. In *ICSLP 2004*.

Vukotic, V., Raymond, C., and Gravier, G. (2015). Is it time to switch to word embedding and recurrent neural networks for spoken language understanding? In *InterSpeech*.