Neural Networks for Spoken Language Understanding

Edwin Simonnet

Université du Maine
École Doctorale STIM
Sciences et Technologies de
l'Information et Mathématiques

Spécialité : Informatique
Laboratoire : LIUM
Equipe : LST

Neural Networks for Spoken Language Understanding

Edwin SIMONNET
Mél : edwin.simonnet@univ-lemans.fr

Abstract: This article explores the use of neural networks, a supervised classification method, to perform a task of spoken language understanding in order to treat automatically user requests over the phone. First the spoken language understanding is defined inside the process chain of automatically treating a user request going from recording the request to giving the appropriate answer. Then are described the several architectures of neural networks used in this study, going from the simplest to the most complex. Some structures allow the network to make past and future prediction in the sentence and other to focus on the important part in a sentence. The results improve significantly as we build more elaborated structure of neural networks. The learning of the neural networks can also be enhanced by bringing additional semantic and syntactic information to the words in the input. Finally we see how the errors made in the other modules involved in the speech processing make the spoken language understanding task more complicated.

Keywords: *Spoken Language Understanding, Neural Networks, Recurrent, Attention Mechanism*
Collaborations : Thanks to the ANR agency for funding through the CHIST-ERA ERA-Net JOKER under the contract number ANR-13-CHR2-0003-05.

1 Introduction

1.1 SLU definition

Spoken Language Understanding (SLU) is usually associated to the automatic extraction and representation of the meaning supported by the words in an uttered sentence [1]. Several modules intervene when a human speaks with a computer. The SLU is one of them. These modules usually are:

1. Automatic Speech Recognition (ASR): to produce the transcription from a speech signal to words.
2. SLU: to extract a semantical representation from words.
3. Treatment and data management: make the appropriate action according to the user request understood *e.g* get an information in a database.
4. Speech synthesis: to give an answer to the user while staying in an oral dialogue.

Nowadays, extracting semantical information out of an open speech remains a difficult task since it is difficult to obtain a generic meaning. In our case the SLU is about phone user requests concerning a specific field which reduces the complexity. Our task consists of treating hotel reservation and touristic information. The semantic representation (what understand the machine) can so be reduced to a set of semantic concept associated to values. To resume, in this context the SLU corresponds to a concept tagging task which is the extraction of a sequence of semantical concept from a sequence of words in order to interpret the meaning of the user request.

This kind of task can be resolved automatically with a supervised classification method among which Neural Networks (NN) which give good results. NN are also used for other tasks as translation [2] which is similar here if we consider we are translating from words (source language) to semantic concepts (target language).

1.2 The MEDIA corpus

The corpus the NN works on is the MEDIA corpus [3]. It is a French corpus about hotel reservation and touristic information which contains dialogues over the phone between users and a simulated automatic system (Wizard of Oz protocol). The aim of this corpus is to train and evaluate the performance of the NN system. Only the user turns are used. Each turn had been manually transcribed and annotated with 74 semantic concept. Working on manual transcription instead of ASR one is useful since it does not add the ASR errors to the SLU errors. The drawback is that it does not place the SLU system in the chain of modules described earlier composed of an ASR system and a SLU system. To be closer to the final task we can also work on ASR transcription and the concepts automatically aligned with them. This however have an impact on the results as we will see later.

A semantic concept can cover several words. Consequently a concept can be a group of tags. Tags are then grouped in concept thanks to B (Begin) I (Inside) suffixes. If a word is the first of a concept, it gets the tag *concept-B*. The following take the tag *concept-I*. It helps the system to easily detect the limits of a concept. A word without semantic information gets the *null* tag. From this concept tagging operation, values associated to concept can also be extracted. Figure 1 show a global example of this.

WORD	je	veux	réserver	une	chambre
CONCEPT	commande			nombre	objet
TAG	commande-B	commande-I	commande-I	nombre-B	objet-B
VALUE	réservation			1	chambre

Figure 1: Example of MEDIA corpus.

The corpus is divided in three part TRAIN, DEV and TEST:

- The TRAIN part is used to train the system. The NN is given both the source (words) and the target (concept tags) and calibrates itself to learn to find them.
- The DEV part is used for validation. At several moment during the training process (on TRAIN) the system performances will be evaluated on DEV. The best calibration of the NN will be kept.
- The TEST part is used after the system has been trained and validated. It is a portion of data the system has never encountered and is used to give a legit evaluation of the system. The results given in this article are on the TEST.

2 Neural Networks description

2.1 Neural Networks

A NN is a set of nodes (neurones) grouped in layers and bound by weighted connexion as seen in figure 2. A node is a real value. The input layer (w) takes a word in a embedded numerical form called word embedding [4]. The NN passes this input layer into one or several hidden layer (h) and then to an output layer (s) which gives probabilities for each possible tag. The most likely is the one kept in the hypothesis. It is the connexion weight which are optimized during the training process to get the more suitable configuration (chosen on DEV).

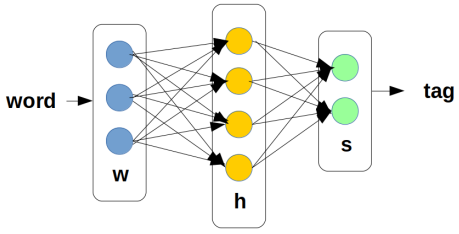

Figure 2: Neural Network Schema

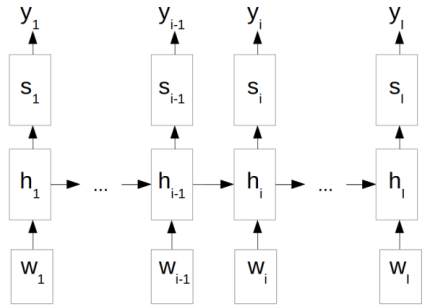

Figure 3: Forward Recurrent Neural Network Schema

2.2 Recurrent Neural Networks

Another way to build NN is with recurrence. A Recurrent Neural Network (RNN) can be forward, backward or bidirectional.

In the case of a forward RNN as seen in figure 3, for a given word, the output of the previous hidden layer is re-injected in the current one. It gives an additional virtual context: the neurones of the hidden layer keep information from previous hidden layers contexts. Thus the RNN can perform sequence prediction beyond the the capacity of a simple feed-forward NN as seen previously where the information is transmitted linearly from the input to the output.

A backward RNN works the same way but taking the sentence backward in order to produce an output based on a future context.

A bidirectional RNN is the combination of a forward and backward RNN both already trained as seen in figure 4. At each position in the sentence we dispose information on both past and future. It gives an important improvement compared to only forward or backward RNN [4].

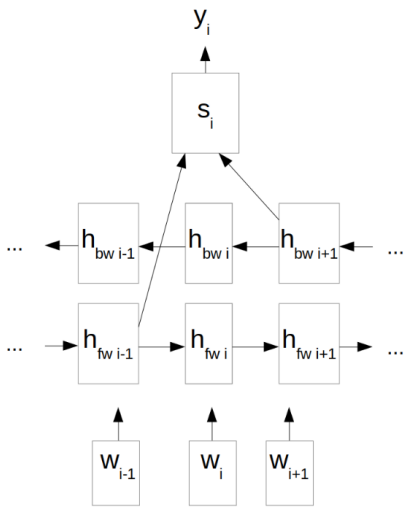

Figure 4: Bidirectional Recurrent Neural Network Schema

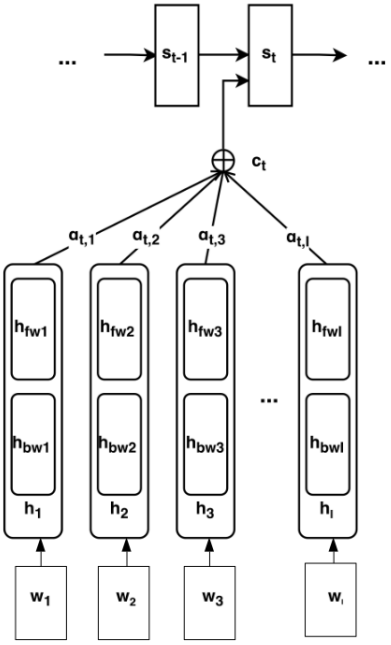

Figure 5: Encoder-Decoder Bidirectional RNN structure with a Mechanism of Attention Schema

2.3 Recurrent Neural Networks with Attention Mechanism

One more improvement brought to the RNN is the attention mechanism to form an encoder-decoder bidirectional RNN structure with a mechanism of attention (BD-RNN-MA) [5] as seen in figure 5.

This BD-RNN-MA computes an annotation h_i for each word w_i from the input sequence w_1 , ... ,w_I . This annotation is the concatenation of the matching forward hidden layer state and the backward hidden layer state obtained respectively by the forward RNN and the backward RNN comprising the bidirectional RNN. Each annotation contains the summaries of the dialogue turn contexts respectively preceding and following a considered word. The sequence of annotations h_1 , ... ,h_I is used by the decoder to compute a context vector c_t (represented as a circle with a cross). A context vector is 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 α_{tj} allows the decoder to choose parts of the input sequence to pay attention to. This context vector is used by the decoder in conjunction with the previous emitted label output y_{t-1} and the current state s_t of the hidden layer of a RNN to make a decision about the current label output y_t.

3 Additional Improvement with Features

3.1 Set of Features

For each words we also dispose information about them that we can add as a supplementary input. These features are the following:

1. its pre-defined semantic categories which belongs to:
   - MEDIA specific categories: like names of the streets, cities or hotels, lists of room equipments, food type, ... *e.g.*: TOWN for Paris
   - more general categories: like figures, days, months, ... *e.g.*: FIGURE for thirty-three.
2. a set of syntactic features: the lemma, part of speech, its word governor and its relation with the current word.
3. a set of morphological features: the 1-to-4 first letter, the 1-to-4 letter last of the word and a binary feature that indicates if the first letter is an upper one.
4. two ASR confidence measures which describe the probability of a ASR word to be wrongly transcribed.

In the experiment on the manual transcription we only disposes of features 1, 4 and part of speech. In the ASR experiment we dispose of all the features.

4 Experimental Results

4.1 Evaluation of the system

The metric used to evaluate the NN on the DEV or the TEST is the Concept Error Rate (CER) measure. When the system produces an hypothesis (an output of semantic tags from words given an input of words), the tags are firstly regroup into concept (thanks to the BI suffixes). Then theses concepts will be compared to the reference one. The CER takes into account the substitutions (S), deletions (D) and insertions (I) between the hypothesis and the reference following the formula $\frac{S+D+I}{N}$ where N is the number of concept in the reference. It is a percentage of error. The lowest it is, the better the system.

4.2 Comparison of the different system configurations on manual transcription

The results in table 1 shows the improvement brought by a bidirectional RNN over forward and backward alone showing how the neural networks performs better with both past and future information. The attention mechanism also bring an improvement other the bidirectional RNN alone.

4.3 Features contribution on manual transcription and ASR

In table 2 we can see how adding features to the input bring another improvement showing that semantic and syntactic information is useful for the system. Finally in table 2 also we can see the same improvement brought by the features but also how passing to the ASR make the task more complicated.

System	CER
Forward RNN	21.2
Backward RNN	19.3
Bidirectional RNN	15.3
BD-RNN-MA	12.9

Table 1: RNN configurations on manual transcription

Features	Transcription	CER
none	manual	12.9
all	manual	11.9
none	ASR	24.2
all	ASR	23.2

Table 2: Feature contribution on BD-RNN-MA system

5 Conclusion

This study aims to explore the use of neural networks in spoken language understanding. Experiments show how we can reach better results by constructing more complex architecture of neural networks allowing them to get information from past and future with the bidirectional RNN but also to focus on the relevant part of an input with the mechanism of attention. Besides we show how bringing additional semantic and syntactic information to the word in the input can improve again the system. Finally results on ASR are shown. It is the goal of SLU to work on ASR but the errors of ASR itself makes the task more complicated. Further improvement are needed to surpass this complexity. Future work is directed on the field of error detection. By using ASR error detection score and by enriching the set of semantic tags with error specific tags, RNN can learn to jointly tag concept and detect ASR error. Other improvement can also be made by using several systems to produce a stronger hypothesis by consensus. Those system can be RNN or other classification system like Conditional Random Field, a state-of-the-art system in SLU.

References

[1] Renato De Mori, Frederic Bechet, Dilek Hakkani-Tür, Michael McTear, Giuseppe Riccardi, and Gokhan Tur. Spoken language understanding. *Signal Processing Magazine, IEEE*, 25(3):50–58, 2008.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Hélene Bonneau-Maynard, Matthieu Quignard, and Alexandre Denis. Media: a semantically annotated corpus of task oriented dialogs in french. *Language Resources and Evaluation*, 43(4):329–354, 2009.

[4] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*, pages 3771–3775, 2013.

[5] Edwin Simonnet, Nathalie Camelin, Paul Deleglise, and Yannick Esteve. Exploring the use of attention-based recurrent neural networks for spoken language understanding. In *NIPS*, 2015.