Spoken Language Understanding (SLU) Generative and Discriminant Models comparison

|  |  |
| --- | --- |
| 000  00  00  00  00  00  00  00  00  00  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |

|  |
| --- |
|  |
| Rodrigo Joni Sestari  number: 179020  rodrigo.sestari@studenti.unitn.it |
|  |
|  |
|  |
|  |
|  |

Abstract

This document has the objective of illustrate the comparison between Generative compiled in the previous document and Discriminant models. This document consists in the second project for the course of Language Understanding System of the University of Trento. This project is available on **github.com/rodrigosestari/LUS.git**

Introduction

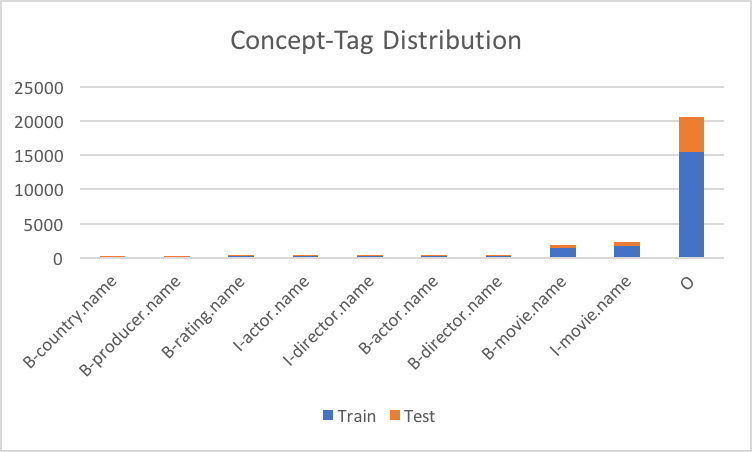
This document describes the steps necessary to implement a Discriminant model using recurrent neural networks and their comparison with Generative model where was used Finite State Transducers. The first part consists in a theory part about Neural Networks, the second one consist to apply the Discriminant model in different parameters, the last part consist to evaluate these models.

Dataset structure

The dataset available to this project are divided by Train and Test:

* NLSPARQL.train.data: Dataset used to train the models. The file is format by the tuple *<word, concept>* the concept use **IOB Notation**. one for each line, and each of these sentences is separated by an empty line
* NLSPARQL.test.data: Dataset used to test, this dataset has the same structure of NLSPARQL.train.data
* **IOB Notation**: *Inside, Outside, Beginning* is a common tagging format for tagging tokens in a chunking task in computational linguistics. The B- prefix before a tag indicates that the tag is the beginning of a chunk, and an I- prefix before a tag indicates that the tag is inside a chunk. The B- tag is used only when a tag is followed by a tag of the same type without O tokens between them. An O tag indicates that a token belongs to no chunk.

Is possible see in the figure bellow the IBO distribution in the dataset:



Theory about natural language process

I this chapter introduces some principles about the natural language processing that come used in this document:

* **N-gram**: n-gram is a contiguous sequence of n items from a given sequence of text or speech. In this document, the sequence come from the Dataset.
* **FSA**: *Finite State Automaton,* is a mathematical model of computation. It is an abstract machine that can be in exactly one of a finite number of states at any given time.
* **FST**: *Finite State Transducer,* is a FSA whose state transitions are labelled with both input and output symbols.
* **WFST**: *Weight Finite State Transducer,* A weighted transducer puts weights on transitions in addition to the input and output symbols.
* **LM**: *Language Model* compute the probability of a sentence or sequence of words:
* **Smoothing**: Assume that no n-gram of known words has 0 probability and redistribute probability mass from seen to unseen events, this is known as smoothing.
* **ANN**: *Artificial Neural Network*, is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems.
* **RNN**: *Recurrent Neural Networ,k* is a class of artificial neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behaviour. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs.

Tools

To execute the tasks about this document was necessary the following tools:

* **OpenFST**: an open-source library for weighted finite-state transducers WFSTs.
* **OpenGRM**: is used for making and modifying n-gram language models encoded as weighted finite-state transducers (FSTs).
* **Conlleval**: A Perl script that allow evaluate the SLU result.
* **Theano:** A Python library for efficiently handling mathematical expressions involving multi-dimensional arrays (also known as tensors). It is a common choice for implementing neural network models. Theano has been developed in University of Montreal, in a group led by Yoshua Bengio, since 2008.

RNN Architecture

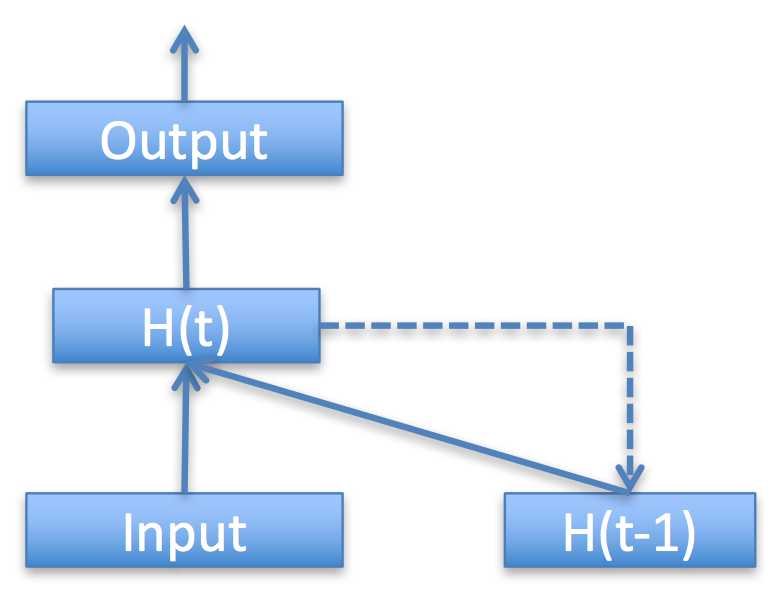
Research on language modelling for speech recognition has increasingly focused on the application of neural net- works. Two competing concepts have been developed: On the one hand, feedforward neural networks representing an n- gram approach, on the other hand recurrent neural networks that may learn context dependencies spanning more than a fixed number of predecessor words.

When a feedforward neural network (FFNN) is used, only the direct predecessor words are used to predict the probability of the current word . Although it is possible to include words from the previous sentence, most of the time the history is truncated at the beginning of the sentence in the n-gram approach. When a recurrent neural network is used, the full sequence of predecessor words is considered for predicting .

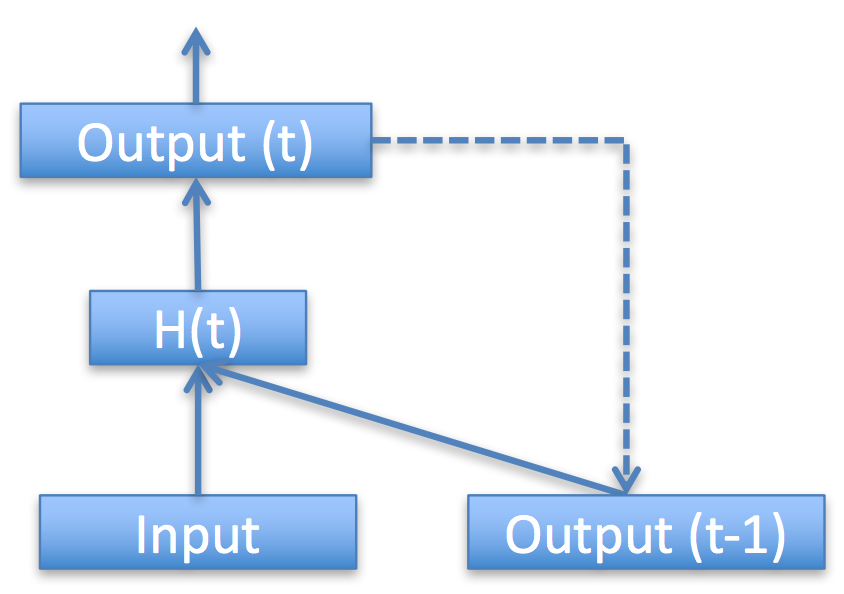
In another way, in a ANN the inputs and outputs are independent between their, the problem is that is not possible to explorer the sequential information. To explorer it, exist the RNN in which, the weighted connections feeding a neuron, also come either from the hidden units of the previous iteration, or from the outputs, been possible consider sequences.

In order to apply a RNN into SLU is used to type of networks.

* **Elman mode**: This network has a connection that feeds the activation of the hidden layer in the previous time step with the current input. The probability of label is estimates by where C is the label, the word an the previous state.



* **Jordan model**: this model connection that feeds the activation of the output layer at previous time step with the current input. The probability of label is estimates by where C is the label, the word an the previous output.



Implementation

The implementation consists to execute some scripts to train and test the LM, these scripts use a external tools Theano.

Scripts

Some scripts are implemented to elaborate chain of input/output files. The scripts are:

* **start.sh**: Used to create the initial training and validation data, and the word and label dictionary. These script requires one parameter, if 0, run a default configuration file, if 1, run for 7 different configuration files.
* **files.sh:** Used in order to create the a the auxiliary dataset **…**

Configuration

The configuration file contains these parameters:

* **lr**: starting learning rate
* **win**: context window size
* **bs**: mini batch size
* **nhidden**: size of the hidden layer
* **seed**: random seed
* **emb\_dimension**: dimension of the word embedding
* **nepochs**: maximum number of backpropagation steps

was created 8 configuration files in order to test all parameters, bellow is possible see these models:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Mo** | **lr** | **win** | **bs** | **hid** | **seed** | **dim** | **epoc** |
| 0 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |
| 1 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |
| 2 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |
| 3 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |
| 4 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |
| 5 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |
| 6 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |
| 7 | 0.1 | 9 | 5 | 100 | 3842845 | 100 | 25 |

Results

|  |  |
| --- | --- |
|  |  |

The idea is measure the output given by the SLU model.

Evaluation Methods

* **accuracy**: Accuracy refers to the closeness of a measured value to a standard or known value.
* **precision**: Precision refers to the closeness of two or more measurements to each other or also called positive predictive value, is the fraction of retrieved instances that are relevant.
* **recall**: recall is the fraction of relevant instances that are retrieved in another way is a measure of how many truly relevant results are returned.
* **FB1**: is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score

Evaluation

In the tables above is possible see the result of evaluation, the tables are divided by smoothing method, the column n-gram represent the n-gram order considered, the next 3 columns represent the evaluation methods; accuracy, precision, recall and F-measure.

**FST**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **method** | **accuracy** | **precision** | **recall** | **FB1** |
| **absolute** | 92.88% | 76.97% | 75.34% | 76.15 |
| **Katz** | 92.62% | 78.03% | 73.88% | 75.89 |
| **Kneser** | 92.90% | 76.79% | 75.53% | 76.16 |
| **Pre-smo** | 92.65% | 78.41% | 74.24% | 76.27 |
| **Unsmoo** | 92.54% | 78.09% | 73.88% | 75.93 |
| **WittenB** | 92.90% | 77.03% | 75.62% | 76.32 |

**RNN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **n-gram** | **accuracy** | **precision** | **recall** | **FB1** |
| 1 | 88.84% | 55.51% | 60.04% | 57.68 |
| 2 | 92.68% | 78.51% | 74.34% | 76.37 |
| 3 | 92.62% | 76.58% | 74.61% | 75.58 |
| 4 | 92.86% | 77.11% | 75.34% | 76.22 |
| 5 | **92.90%** | 77.03% | 75.62% | 76.32 |

Comparison

After train test and evaluate all methods of smoothing is possible get some conclusions.

Is evident that the accuracy is equal for all smoothing methods with unigram, the motive can be the fact that unigram use the word frequency so the probability becomes the same for al methods. The method with the worse accuracy median was Pre-smoothing, instead Witten-bell with 5 n-grams was the best with 92.90% of accuracy. Is important observe that the performance difference between the methods with the same n-gram order is very small. The accuracy with cut-off threshold positive was worse for all smoothing methods than without cut-off, the file results can be seen on [*https://github.com/rodrigosestari/LUS*](https://github.com/rodrigosestari/LUS) */blob/master/First/result/*

References

John Wiley and Sons. New York, 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*

Maccartney, B. *(2005).* Stanford.from *https://nlp.stanford.edu/~wcmac/papers/20050421-smoothing-tutorial.pdf*

Mohri et al. (1996) *FSM Toolkit Exercises*

Riccardi (2017). UniTN. Retrieved 10 April, 2017, from http://disi.unitn.it/~riccardi/