Contribution to Semantic Parsing Approaches and Techniques for Natural Language Processing

Gregorio Nuevo Castro
Supervised by: Felipe Gil Castiñeira[1]

[1]Department of Telematics Engineering, University of Vigo

Introduction

Semantic Parsing is a topic of *Natural Language Processing* (NLP) that aims to extract interesting meaning from text in the form of relations between words (figure 1).

Figure 1 : Semantic parsing.

Artificial **Neural Networks** have recently contributed to different research fields, with the development of better models with many layers (the so called *Deep Learning*). They can learn complex representations of the input (see figure 2).

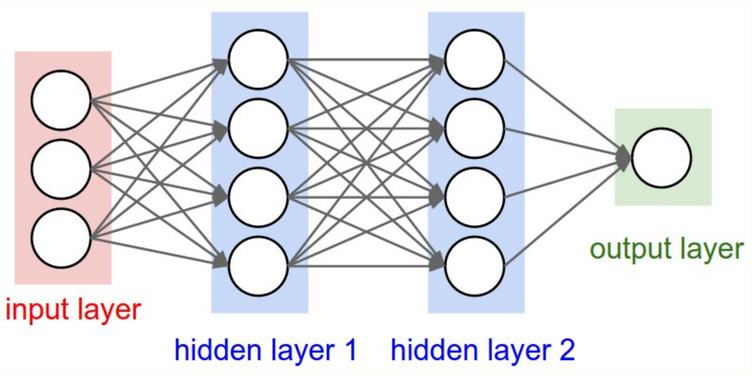


Figure 2 : A neural network.

NLP have been greatly benefited with the use of neural nets. Most of this success have been provided by the use of *word embeddings*.

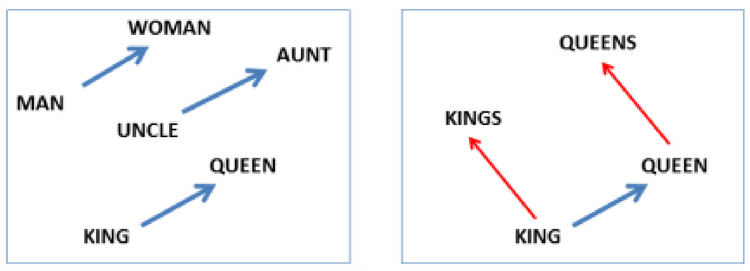


Figure 3 : Word embeddings.

Word embeddings (e.g. *word2vec* [3] and *GloVe* [4]) are vectorial representations that capture the *syntactic/semantic* meaning of a word in a high dimensional space. Some meaningful relations, such as (for example) *gender* or *number*, can be established between words, as shown in figure 3. Embeddings have the ability of ordering words in its space and are of great utility as input for NLP problems.

Motivation

NLP processing is a promising field in the intersection between *linguistics*, *machine learning* and *artificial intelligence*. Research in semantic parsing will help not only to solve other NLP tasks: *sentiment analysis*, *question answering*, *named entity recognition*, etc. ; but also real world problems:

- **Information Retrieval** (IR) from documents;
- **Question answering** (QA);
- **Human Computer Interaction** (HCI), conversational bots, intelligent assistants (i.e. Siri, Cortana, Google Now), etc.

Objectives

Contribute to the research in **Semantic Parsing**. Despite of the great advances in semantics, there is still no NLP system able to deal with semantic information in a general free-context manner, and therefore boosting our knowledge from **Natural Language Processing** to **Natural Language Understanding** (NLU), see figure 4.

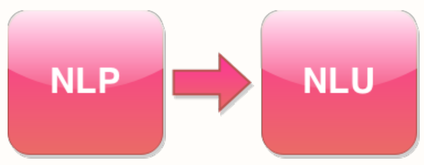


Figure 4 : From Natural Language Processing (NLP) to Natural Language Understanding (NLU).

Research Plan

Current State

Experimenting with computing word embeddings from their morpheme segmentation (figure 5), in a similar manner than [2]. Based on this research, we are writing a paper to explain the system we are building and its results.

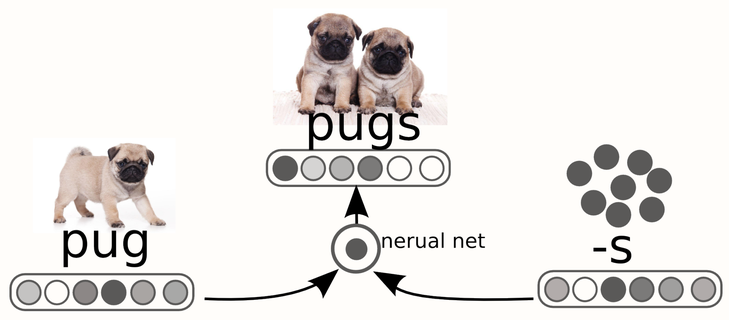


Figure 5 : Composing an embedding for *pugs*, from an embedding for *pug* and *-s*.

Expected results

We expect to get

- an LSTM trained model that
- accurately generates word embeddings from morpheme embeddings; thus
- dealing with both normal and strange words; and
- reducing the embedding dictionary to a minimum (figure 6).

Figure 6 : *Morphemes dictionary – words dictionary* comparison.

Resources

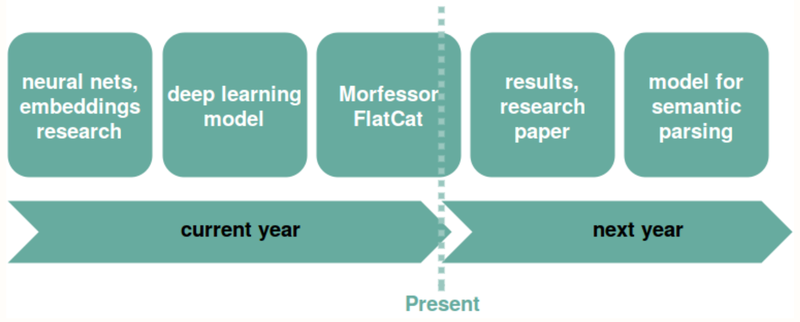
We use several resources for our research. We would like to highlight two of them: *Theano* and *Morfessor FlatCat*.

Figure 7 : Theano logo.

Theano [5] is a Python library to define, optimize, and evaluate mathematical expressions. It is used to efficiently compute multi-dimensional array operations, needed to train and run a deep neural net.

Morfessor FlatCat [1] is a tool and a series of methods to extract mophological segmentation of words from natural language input.

Next year planning



References

[1] S.-A. Grönroos, S. Virpioja, P. Smit, and M. Kurimo. Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *COLING*, pages 1177–1185, 2014.

[2] T. Luong, R. Socher, and C. D. Manning. Better word representations with recursive neural networks for morphology. In *CoNLL*, pages 104–113. Citeseer, 2013.

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

[4] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[5] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.