Is it time to switch to Word Embedding and Recurrent Neural Networks for Spoken Language Understanding?

Vedran Vukotic, Christian Raymond, Guillaume Gravier

Is it time to switch to Word Embedding and Recurrent Neural Networks for Spoken Language Understanding?

Vedran Vukotic[1,2]*, Christian Raymond*[1,2], Guillaume Gravier[2,3]

[1]INSA de Rennes, Rennes, France
[2]INRIA/IRISA, Rennes, France
[3]CNRS, France

{vedran.vukotic,christian.raymond,guillaume.gravier}@irisa.fr

Abstract

Recently, word embedding representations have been investigated for slot filling in Spoken Language Understanding, along with the use of Neural Networks as classifiers. Neural Networks, especially Recurrent Neural Networks, that are specifically adapted to sequence labeling problems, have been applied successfully on the popular ATIS database. In this work, we make a comparison of this kind of models with the previously state-of-the-art Conditional Random Fields (CRF) classifier on a more challenging SLU database. We show that, despite efficient word representations used within these Neural Networks, their ability to process sequences is still significantly lower than for CRF, while also having a drawback of higher computational costs, and that the ability of CRF to model output label dependencies is crucial for SLU.

Index Terms: spoken language understanding, word embedding, CRF, neural network

1. Introduction

The focus of this paper is in Spoken Language Understanding (SLU). In classical SLU systems, one of the key tasks is to label words with lexical semantics. For example, in the sentence "I want a Chinese restaurant near Tour-Eiffel", the word "Chinese" should be labeled as the food-type of a restaurant, and "Tour-Eiffel" as a relative place in Paris.

Many sequence labeling methods have been investigated in SLU: SVM [1], HVS [2], Machine translation models, Finite State Transducers and particularly Conditional Random Fields, which have been shown in [3] to be best-suited for this task. Recently, Neural Networks have been investigated in [4, 5] where they show, on the popular ATIS database, that Recurrent Neural Networks and Long Short Term Memory Neural Networks provide state-of-the-art results. Nevertheless, a wide variety of methods are able to provide very good results on ATIS [6], including methods that are not dedicated to sequence labeling (*e.g.* SVM). These last methods fail [7, 3] when evaluated on MEDIA [8], another SLU database. This indicates that ATIS is not very challenging and conclusions obtained on this database are not particularly strong. In this paper we propose to evaluate some of the popular Neural Networks on the SLU concept tagging task on two different databases, namely ATIS and MEDIA and to compare them to Conditional Random Fields [9], the previous state-of-the-art method on these two corpuses [6, 3].

Recent Neural Networks come together with new text representations where the symbolic text representation is mapped to a numeric one using popular word embedding methods [10, 11]. This representation has several advantages, the most salient one is to make words that are syntactically or semantically related close to each-other in the representation space. One question that arises is to know whether improvements come from the representation, the classifier itself or maybe both. However, for SLU, a precise word clustering is already available: the attribute database linked to the task (*e.g.* city names, airline names for ATIS, etc.), making this advantage not clear in the case of SLU. We thus propose to compare both of them under the same classification algorithm, in order to make a strict comparison.

The paper is structured as follow, first we will present the two databases, ATIS and MEDIA, used for our evaluations in Section 2. Symbolic and numeric word representations for SLU are compared in Section 3. In Section 4 we will compare the recently proposed Neural Networks [4] against CRF on the two databases. We will show that CRF still significantly outperforms Neural Networks on the MEDIA database in terms of accuracy, rapidity and flexibility.

2. Datasets

In our experiments we used two datasets: ATIS and MEDIA. ATIS is a publicly available corpus used in the early nineties for SLU evaluation. MEDIA has been collected in the last decade and is available through ELRA since 2008.

2.1. ATIS

The Air Travel Information System (ATIS) task [12] is dedicated to provide flight information. The semantic representation used is frame based. The SLU goal is to find the good frame and fill the corresponding slots.

$$\begin{bmatrix} \text{words: flights from boston to philadelphia} \\ \text{FRAME:} \quad \text{FLIGHT} \\ \qquad\quad \text{DEPARTURE.CITY = boston} \\ \qquad\quad \text{ARRIVAL.CITY = philadelphia} \end{bmatrix}$$

The training set consists of 4978 utterances selected from the Class A (context independent) training data in the ATIS-2 and ATIS-3 corpora while the ATIS test set contains both the ATIS-3 NOV93 and DEC94 datasets.

2.2. MEDIA

The research project MEDIA [8] evaluates different SLU models of spoken dialogue systems dedicated to provide tourist information. A 1250 French dialogue corpus has been recorded by ELDA following a Wizard of Oz protocol: 250 speakers have each followed 5 hotel reservation scenarios. This corpus has been manually transcribed, then conceptually annotated accord-

ing to a semantic representation defined within the project. This representation is based on the definition of concepts that can be associated with 3 kinds of information. First a concept is defined by a label and a value; for example with the concept date, the value 2006/04/02 can be associated. Second, a specifier can be attached to a concept in order to link the concept, in order to go from a flat concept/value representation to a hierarchical one; for example, the concept date can be associated with the specifiers *reservation* and *begin* to specify that this date is the beginning date of a hotel reservation. Third, modal information is added to each concept (positive, affirmative, interrogative or optional). Table 1 shows an example message from the MEDIA corpus with only concept-value information. The first column contains the segment identifier in the message, the second column shows the chunks W^c supporting the concept c of the third column. In the fourth column the value of the concept c in the chunk W^c is displayed. The MEDIA semantic dictionary contains 83 concept labels, 19 specifiers and 4 types of modal information. In this study we will focus only on concept extraction. No specifiers, values or modal information are considered, so the tagset considered consists solely of 83 labels. The MEDIA corpus is split into 3 parts. The first part (720 dialogues, 12K messages) is used for training the models, the second (79 dialogues, 1.3K message) when cross-validation is performed, and the third part (200 dialogues, 3.4K message) is used as test.

3. Symbolic vs embedded

For concept labeling in SLU, features commonly consist of word observations associated with their relative position from the decision point in the sequence. For symbolic representations, the feature set is then a bag of pairs "word/relative position" within a specific sliding window of observation. For numeric representations, the feature set is obtained by word embedding methods [10, 11]. The final vector is a concatenation of the numeric representation of each word that belong to this sliding windows. A common window of $[-2, +2]$ [6, 3] or $[-3, +3]$ [4, 13] is generally sufficient to obtain satisfactory performances. In this work we opted to use the latter for performing the comparison, although different sizes were tested, as mentioned in Section 4.2

As mentioned earlier, in human-machine applications, we have database attributes available to make a fine clustering of many words supporting concepts related to understanding: the list of airline names or city names in ATIS or the list of food type for a restaurant, the list of facilities for a hotel, the list of French cities, *etc.* in MEDIA. These information are added to the set of symbolic features.

To produce a numeric representation from the symbolic ones, we just replace words from utterances by the cluster from

n	W^c	c	**value**
1	yes	answer	yes
2	the	RefLink	singular
3	hotel	BDObject	hotel
4	which	null	
5	price	object	payment-amount
6	is below	comparative-payment	below
7	fifty five	payment-amount-int	55
8	euros	payment-currency	euro

Table 1: Example of message with concept+value information. The original French transcription is: "*oui l'hôtel dont le prix est inférieur à cinquante cinq euros*"

Representation	Precision	Recall	F-measure
ATIS			
symbolic	93.00%	93.43%	93.21%
numeric	93.50%	94.54%	94.02%
MEDIA			
symbolic	71.09%	75.48 %	73.22%
numeric	73.61%	78.85%	76.14%

Table 2: Slot tagging performance obtained from symbolic and numeric representations using bonzaiboost on ATIS and MEDIA

where they belong (*e.g.* city_name, food, *etc.*) and keep the word if it does not belong to any of them. Then we use the word2vector [10] tool to produce the embedding of each token by training only on the training set of the SLU corpus.

The two different representations are then used as input for a classifier that is able to work with both of them, in order to have a strict comparison. We use boosting over decision trees [14]. This algorithm is not specifically designed to work on sequence problems, but the goal is solely to compare the representations. The results are presented in table 2 and they clearly show, on both datasets, that numeric representations improve the accuracy of the classifier. Moreover, we can observe in figure 1 that numeric representations allow the classifier to converge significantly faster than with symbolic representations, on both datasets. The classifier built on ATIS exhibits several drops in accuracy, as it can be seen in figure 1a. Our explanation is that there are annotation errors in the ATIS dataset and each drop corresponds to a rule created from this error by the classifier. As we can see, the numeric representation learned on the same corpus does not suffer from this drawback and appears to be noise robust. Annotation errors in ATIS are known since [6] who proposed a partially corrected ATIS version of the corpus, but some errors still remain [15]. [6] show that in the previous noisy version, a basic HMM worked better than CRF because of their noise resistance ability. After correction, every method benefited and gained up to 5% absolute in accuracy, making CRF the best method. This result indicates that the good results obtained on ATIS by different Neural Network architectures [4, 13, 5] are partially due to the representation itself.

To conclude, it appears that using numeric representations clearly bring advantages compared to using symbolic ones, even if good base clustering is already available from database attributes. This advantage is due to the fact that numerical representations appear less sensitive to noise, avoiding the possibility for the classifier to build a very specific (and false) classification rule.

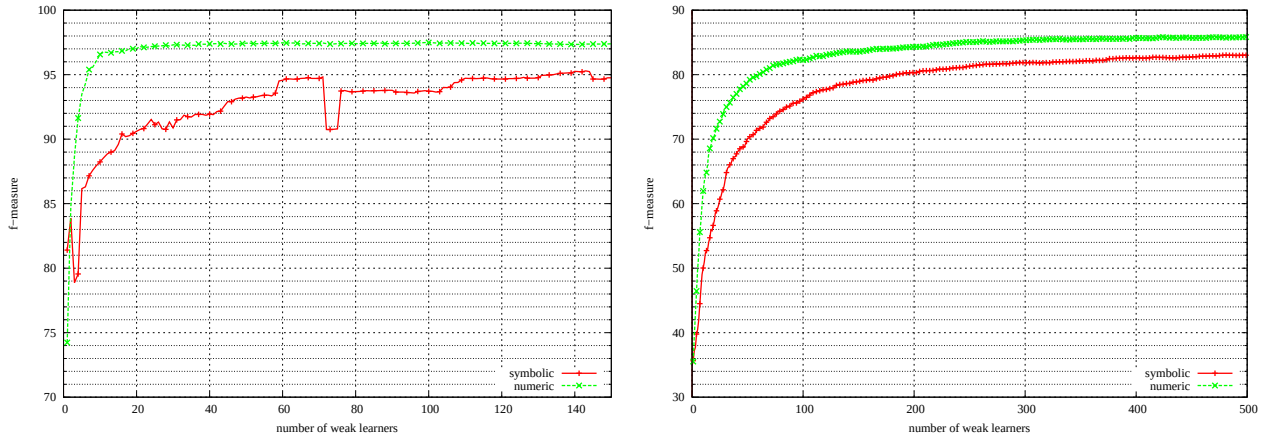
4. Sequence labeling algorithms

4.1. Algorithms

In this section, we compare 3 algorithms: state-of-the-art [6, 3] Conditional Random Fields [9], the recently proposed Elman Recurrent Neural Network as well as the Jordan RNN [4, 13] and the popular AdaBoost.MH [16] over bonsai trees [14].

Each of these algorithms is able to take as input an arbitrary set of features and observe features from preceding and following positions of the sequence in an arbitrary window size. The main differences are described next:

- AdaBoost.MH over bonsai trees is a widely used classi-



(a) ATIS (b) MEDIA

Figure 1: F-measure[1] according to the number of boosting iterations with symbolic and numeric features

fication algorithm that gives in general very good performances on many different tasks. However, it is not dedicated at all to sequence labelling problems. Sequence tagging is done by successive and independent local decisions at each sequence position. Thus, this algorithm will give us a baseline to see improvements brought by the two next sequence adapted classification algorithms. We use the implementation described in [17].

- The standard behavior of a Feedforward Neural Network is the same as for the previous algorithm: a succession of independent and local decisions. In an RNN a recurrence is added to allow the Neural Network to exhibit dynamic temporal behavior. In [4], they use the output of Neural Network from the previous or future time step as a feature for the current NN in the sequence. They proposed to use the hard predicted output or the output probabilities and test these solutions in both directions. In [13] they use as features in their RNN the output of the Hidden Layer of the previous time step. Despite these heuristics to trade off context information along the successive decision, no dependencies on target labels are explicitly modeled and no global decision is made. The RNN architectures tested are an Elman RNN and a Jordan RNN, both proposed by [4]. They have distributed their code based on the Theano library [18, 19].
- A Conditional Random Field, unlike the previous algorithms is dedicated to sequence labelling. Target label dependencies are modeled under the Markov assumption (in order to remain tractable) and then a global decision on the sequence is made. However, popular and efficient implementations like the one we used [20] are capable of using solely symbolic features.

4.2. Features and configuration

All features have been extracted in windows of size $[-3, 3]$ (three words before and three words after the current word). This is a commonly used configuration that also gives the best results for both ATIS and MEDIA. Further increasing the window size didn't affect the result significantly. Smaller context window sizes would however decrease the performance.

For the symbolic feature representation, the feature set is composed of a bag of word/position pairs inside the windows. In case a word is found within the database of attributes (*e.g.* city_name), it is replaced with its corresponding entry prior to computing the representation.

To build the numeric representations, we used the word2vec model [10] trained on the training corpus where words belonging to an attribute database were replaced by their corresponding attribute, in order to transfer this knowledge to the numeric representations. Only one embedding strategy is considered, since when fine-tuned, different word representations show very similar performances and provide comparable results [21]. This is also significantly cost-effective since just a few minutes are sufficient to compute the representations.

Representations in a 100-dimensional space yielded very good results for all the tested classification algorithms. Further increasing the representation dimensionality did not result in a noticeable improvement of the results. This is the size we keep to do the algorithm comparison.

In the RNN implementation [4] word embeddings are learned jointly with the final supervised task-specific classifier (RNN), in order to fine-tune them on the final task. This has a small impact also on the speed of the overall training procedure. Database attributes have been integrated in order to provide a fair comparison.

RNNs have many crucial hyperparameters. We kept most of them fixed to the values proposed in [13]. We ran a 50 epochs learning and the best RNN configuration was selected according to its performances on the development set. On the other side, we kept the default parameters of wapiti. Bonzaiboost was ran with decision trees of depth 2 (max 4 leaves) according to [14]. For ATIS, we used the best data split reported in [13] while for MEDIA, the official split of the dataset has been used.

4.3. Algorithms comparison

The three algorithms were ran on the slot extraction task for both databases: ATIS and MEDIA. Boosting and CRF implementations are multithreaded and were ran with 16 threads on a 2 Intel(R) Xeon(R) CPU X5560 @2.80GHz machine with 96 GB of RAM. The RNN GPU implementation was ran on an

[1] Reported by the classifier, not by conlleval (no sequence evaluation but target label evaluation).

Algorithm	Parameter	Representation	Precision	Recall	F-measure	Training Time
ATIS						
Bonzaiboost	100 iter	numeric (word2vec)	93.50%	94.54%	94.02%	~20 m
Bonzaiboost	100 iter	symbolic	93.12%	92.82%	92.97%	~3 m
CRF		symbolic	95.53%	94.92%	95.23%	~6 m
Elman RNN	100 hdn	numeric (joint)	96.20%	96.12%	96.16%	~1.5h
MEDIA						
Bonzaiboost	500 iter.	numeric (word2vec)	73.61%	78.85%	76.14%	~2.5 h
Bonzaiboost	500 iter.	symbolic	71.09%	75.48 %	73.22%	~34 m
CRF		**symbolic**	**87.70%**	**84.35%**	**86.00%**	**~15 m**
Elman RNN	500 hdn	numeric (joint)	83.36%	80.22%	81.76%	~31 h
Elman RNN	500 hdn	numeric (word2vec)	80.48%	83.46%	81.94%	~22 h
Jordan RNN	500 hdn	numeric (joint)	82.76%	83.75%	83.25%	~3.5 h
Jordan RNN	500 hdn	numeric (word2vec)	83.40%	82.90%	83.15%	~3 h

Table 3: Slot tagging performance obtained with several learning algorithms on ATIS and MEDIA. hdn stands for hidden neurons.

NVIDIA GeForce GT 750M 2048 MB graphic card.

Performances were computed in terms of accuracy, precision, recall and F-measure, using the conlleval script[2]. Training times are also reported as a vague indicator of the complexities of the tested algorithms.

Computations were made with different number of iterations (and hidden neurons for the case of RNNs) to ensure that the asimptote of the learning curve was reached.

Table 3 reports these information for both ATIS and MEDIA.

4.3.1. ATIS

As it can be seen in table 3, the performances of all the classifiers are very similar: from ~93% in F-measure for bonzaiboost (not dedicated to sequence labeling tasks and applied on symbolic representations) to ~96% for RNN. On the numeric representation, the gap between bonzaiboost and RNN is reduced to 2% absolute only. This result illustrates the fact that ATIS is not particularly challenging in terms of sequence classification. RNNs perform better (~1% absolute) than CRF on ATIS. As pointed out in the previous Section 3, the representation used (symbolic for CRF and numeric for RNN) may explain the RNN gain. This result is also pointed out by the authors of [13].

4.3.2. MEDIA

On MEDIA, results are substantially different for each classifier. As expected, bonzaiboost, which is not dedicated at all to sequence labeling, produced the worst performance, around 76%. RNNs follow with 83.25% at the cost of high computational time. CRF, despite the fact that it is using less efficient symbolic representations, obtains 86% with less computational cost (15min *vs* 3.5h).

The Jordan variation of RNNs shows a less stable convergence. Elman RNNs had quite more stable convergences. Word embeddings learned in an unsupervised manner (word2vec) combined with an RNN perform similarly to word embeddings computed in a supervised manner, while learning the RNN classifier. However, precomputing the embeddings significantly decreases the time required for training an RNN classifier and helps the classifier to converge faster.

On the formulation side, CRF has the advantage to model explicitly the dependencies between target labels. To keep the CRF tractable, the linear chain CRF is widely used. This means that only dependencies between two adjacent labels are modeled. If we remove features related to these dependencies, CRF loses ~6% absolute in terms of F-measure. This result clearly indicates that the good CRF performances derive from this dependency model.

[2]http://www.cnts.ua.ac.be/conll2000/chunking/output.html

5. Conclusion

We compared symbolic and numeric word representations for SLU with a classification algorithm able to use both. Our results demonstrate that the latter allows a better generalization (better accuracy) and make the classification algorithm to converge faster. Moreover, numeric representations decrease the possibility for a classifier to produce noise fitted decision rules and thus are more robust to noise than symbolic ones. Despite this conclusion, algorithms able to exploit them, like RNNs are not able to compete with CRF. Although CRF is trained solely on symbolic features, its ability to model output label dependencies appears crucial for the task. CRF with symbolic features thus remains the best classification algorithm for SLU, in term of prediction (2.75% absolute gain of F-measure in the challenging MEDIA corpus and a 16% relative decrease of the error), simplicity (less hyperparameters) and rapidity (~14 times faster in our experiments).

6. References

[1] T. Kudo and Y. Matsumoto, "Chunking with Support Vector Machines," in *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, ser. NAACL '01. Stroudsburg, PA, USA: Association for Computational Linguistics, 2001, pp. 1–8. [Online]. Available: http://dx.doi.org/10.3115/1073336.1073361

[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] 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 Transactions on Audio, Speech and Language Processing*, vol. 19, no. 6, pp. 1569–1583, August 2011.

[4] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication*

Association, Lyon, France, August 25-29, 2013, 2013, pp. 3771–3775. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2013/i13_3771.html

[5] K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, and Y. Shi, "Spoken language understanding using long short-term memory neural networks," *IEEE Spoken Language Technology Workshop*, 2014.

[6] C. Raymond and G. Riccardi, "Generative and Discriminative Algorithms for Spoken Language Understanding," in *InterSpeech*, Antwerp, Belgium, August 2007, pp. 1605–1608.

[7] S. Hahn, P. Lehnen, C. Raymond, and H. Ney, "A comparison of various methods for concept tagging for spoken language understanding," in *Proceedings of the Language Resources and Evaluation Conference*, Marrakech, Morocco, May 2008.

[8] H. Bonneau-Maynard, S. Rosset, C. Ayache, A. Kuhn, and D. Mostefa, "Semantic Annotation of the French Media Dialog Corpus," in *InterSpeech*, Lisbon, September 2005.

[9] J. D. Lafferty, A. Mccallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *International Conference on Machine Learning*, San Francisco, CA, USA, 2001, pp. 282–289.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *International Conference on Learning Representations*, 2013.

[11] "Word Embeddings through Hellinger PCA, author = R. Lebret and R. Collobert," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics, 2014, pp. 482–490.

[12] D. A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg, "Expanding the scope of the ATIS task: the ATIS-3 corpus," in *HLT*, 1994, pp. 43–48.

[13] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent Neural Networks for Language Understanding," in *InterSpeech*. Interspeech, August 2013. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=200236

[14] A. Laurent, N. Camelin, and C. Raymond, "Boosting bonsai trees for efficient features combination : application to speaker role identification," in *InterSpeech*, Singapour, September 2014.

[15] G. Tur, D. Hakkani-Tur, and L. Heck, "What is left to be understood in ATIS?" in *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE, 2010, pp. 19–24.

[16] R. E. Schapire and Y. Singer, "BoosTexter: A boosting-based system for text Categorization," *Machine Learning*, vol. 39, pp. 135–168, 2000.

[17] C. Raymond. (2013) Bonzaiboost. [Online]. Available: http://bonzaiboost.gforge.inria.fr

[18] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," in *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2012.

[19] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for scientific computing conference (SciPy)*, vol. 4. Austin, TX, 2010, p. 3.

[20] T. Lavergne, O. Cappé, and F. Yvon, "Practical Very Large Scale CRFs," in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, July 2010, pp. 504–513. [Online]. Available: http://www.aclweb.org/anthology/P10-1052

[21] R. Lebret, J. Legrand, and R. Collobert, "Is Deep Learning Really Necessary for Word Embeddings?" Idiap, Tech. Rep., 2013.