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

Vedran Vukotic, Christian Raymond, Guillaume Gravier

presented by

Frédéric Béchet

IRISA/INRIA/INSA, Rennes, France

september 2^{nd}, 2015

Introduction

Spoken Language Understanding

- previously state-of-the-art were Conditional Random Fields [Hahn et al., 2011]
- recently Recurrent Neural Networks brings improvements on the ATIS database [Mesnil et al., 2013]

Introduction

Spoken Language Understanding

- previously state-of-the-art were Conditional Random Fields [Hahn et al., 2011]
- recently Recurrent Neural Networks brings improvements on the ATIS database [Mesnil et al., 2013]

questions:

- where does the RNN gain come from?
  1. classifier ?
  2. representation ?
- are RNNs a better choice for SLU?
  - is the dataset challenging enough to differentiate the two methods?

Possible gain sources

input representation

- symbolic input
- numerical input / embedding

$\hookrightarrow$ compare both inputs with a single independent classifier that can work with both input types

classification algorithm

$\hookrightarrow$ compare the two classifiers on a challenging dataset (MEDIA)

ATIS & Media presentation

ATIS: obtain air travel information such as flight schedules, fares, and ground transportation from a relational database

x= *list* twa *flights from* washington *to* philadelphia

y=<null><airline> <null> <depart.city><null><arrive.city>

MEDIA: reservation of hotel rooms with tourist information.

x=*euh* une chambre pour deux personnes au novotel

y=<null><number> <room-type> <hotel-mark>

ATIS & Media sets

Air Travel Information System

- Train corpus: 4978 utterances
- Test corpus: 893 utterances
- 572 words, 64 labels
- words supporting concept 49%
  - segmentation: easy: almost one word to concept correspondence
  - classification: easy: main ambiguity → departure *vs* arrival info

Media

- Train corpus: 12922 utterances
- Test corpus: 4772 utterances
- 2460 words, 75 labels
- words supporting concept 72%
  - segmentation: hard
  - classification: hard: hierarchical attributes, complex dependencies

ATIS & Media in the literature

ATIS

- best error rate: $\sim$ 4/5%
- many classifiers performs well (8%$\rightarrow$ 4%)

MEDIA

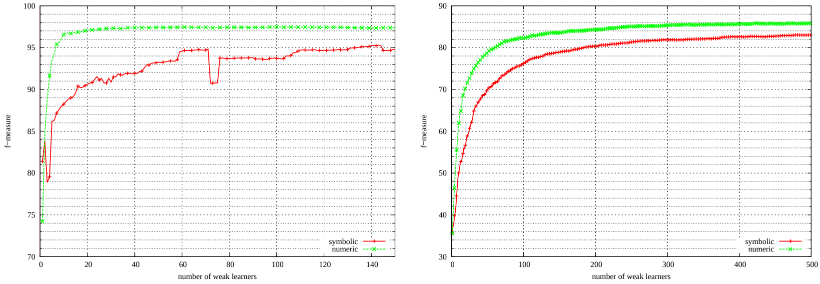
- best error rate: $\sim$ 12%
- CRF perform the best

Symbolic vs embedded inputs

- bonzaiboost (boosting over decision trees) - straight-forward use with both representations
- context window of [-3, 3] words/classes

Symbolic vs embedded inputs

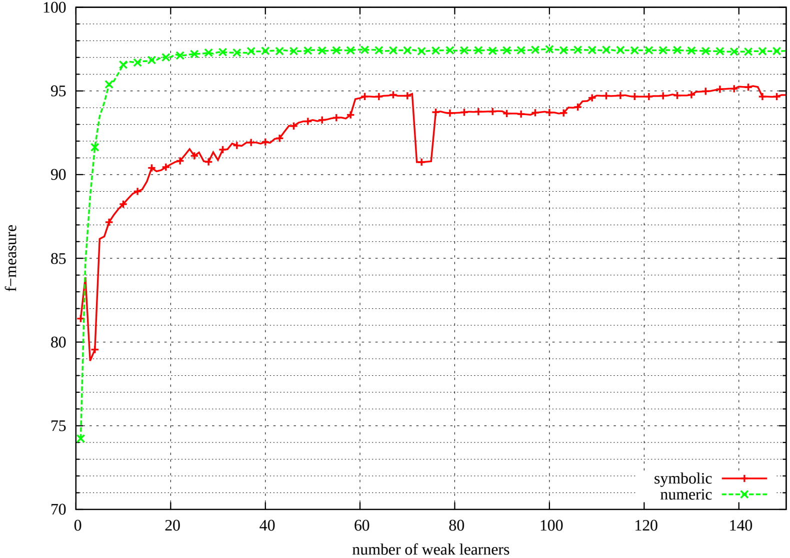
- bonzaiboost (boosting over decision trees) - straight-forward use with both representations
- context window of [-3, 3] words/classes



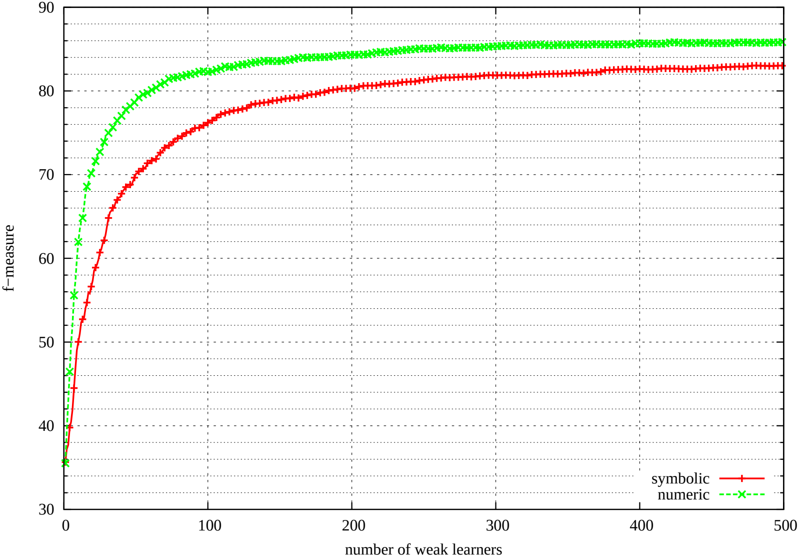
(a) ATIS (b) MEDIA

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

Symbolic vs embedded inputs on ATIS

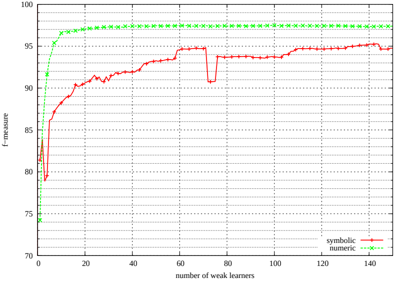


Symbolic vs embedded inputs on MEDIA

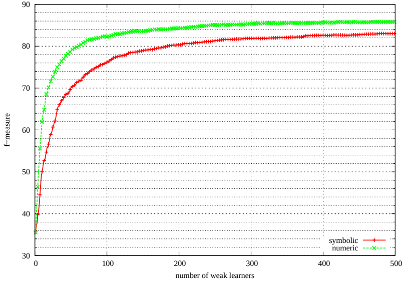


Symbolic vs embedded inputs

- bonzaiboost (boosting over decision trees) - straight-forward use with both representations
- context window of [-3, 3] words/classes



(a) ATIS



(b) MEDIA

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

Symbolic vs embedded inputs

- embedding improves results and convergence speed
  - ATIS: $\sim$ +1%
  - MEDIA:$\sim$ +3%
- robustness to noise (annotation errors)

Symbolic vs embedded inputs

- embedding improves results and convergence speed
  - ATIS: $\sim$ +1%
  - MEDIA:$\sim$ +3%
- robustness to noise (annotation errors)

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%**

Classifiers comparison

- boosting over decision trees
  - not dedicated to sequence labeling: baseline
  - bonzaiboost `http://bonzaiboost.gforge.inria.fr/` [Laurent et al., 2014]
- CRFs
  - dedicated to sequence labeling
  - Wapiti `https://wapiti.limsi.fr/` [Lavergne et al., 2010]
- RNNs
  - Elman Architecture
  - Jordan Architecture
  - supervised (joint) *v.s.* unsupervised(word2vec) embedding
  - public implementation based on Theano `http://deeplearning.net/tutorial/rnnslu.html`

Classifiers comparison: ATIS

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	default	symbolic	95.53%	94.92%	95.23%	**~6 m**
Elman RNN	**100 hdn**	**numeric (joint)**	**96.20%**	**96.12%**	**96.16%**	~1.5h

Classifiers comparison: ATIS

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	default	symbolic	95.53%	94.92%	95.23%	**~6 m**
Elman RNN	**100 hdn**	**numeric (joint)**	**96.20%**	**96.12%**	**96.16%**	~1.5h

- very similar performances
- RNN performs better (~1%)
  - main reason: embedding

Classifiers comparison: MEDIA

Algorithm	Parameter	Representation	Precision	Recall	F-measure	Training Time
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	default	**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

Classifiers comparison: MEDIA

Algorithm	Parameter	Representation	Precision	Recall	F-measure	Training Time
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	default	**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

- CRF obtains best results $\sim$ +3%
  - despite not using embeddings
- Jordan RNN had a less stable convergence
- embeddings learned in a supervised and in an unsupervised manner behave similarly

Conclusion

1. embedding brings improvement
   - even with the presence of word classes knowledge (like city-names, *etc.*)
   - more robust to noise

Conclusion

1. embedding brings improvement
    - even with the presence of word classes knowledge (like city-names, *etc.*)
    - more robust to noise
2. on the (easier) ATIS dataset, performances are very similar
   $\hookrightarrow$ RNNs slightly better thanks to the representation

Conclusion

1. embedding brings improvement
   - even with the presence of word classes knowledge (like city-names, *etc.*)
   - more robust to noise
2. on the (easier) ATIS dataset, performances are very similar
   $\hookrightarrow$ RNNs slightly better thanks to the representation
3. on the (more challenging) MEDIA dataset, CRFs still outperform RNNs
   $\hookrightarrow$+3%

Conclusion

1. embedding brings improvement
   - even with the presence of word classes knowledge (like city-names, *etc.*)
   - more robust to noise
2. on the (easier) ATIS dataset, performances are very similar
   $\hookrightarrow$ RNNs slightly better thanks to the representation
3. on the (more challenging) MEDIA dataset, CRFs still outperform RNNs
   $\hookrightarrow$+3%
4. output label dependencies appear to be crucial
   - CRF $\downarrow$ 6% without them
     $\hookrightarrow$the recurrence in RNN does not model these dependencies efficiently

Conclusion

1. embedding brings improvement
   - even with the presence of word classes knowledge (like city-names, *etc.*)
   - more robust to noise
2. on the (easier) ATIS dataset, performances are very similar
   $\hookrightarrow$ RNNs slightly better thanks to the representation
3. on the (more challenging) MEDIA dataset, CRFs still outperform RNNs
   $\hookrightarrow$+3%
4. output label dependencies appear to be crucial
   - CRF $\downarrow$ 6% without them
     $\hookrightarrow$the recurrence in RNN does not model these dependencies efficiently
5. CRFs are faster and easier to train than RNNs

Hahn, S., Dinarelli, M., Raymond, C., Lefèvre, 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.

Laurent, A., Camelin, N., and Raymond, C. (2014).
Boosting bonsai trees for efficient features combination : application to speaker role identification.
In *InterSpeech*, Singapour.

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.

Mesnil, G., He, X., Deng, L., and Bengio, Y. (2013).
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*, pages 3771–3775.