Adapting lexical representation and OOV handling from written to spoken language with word embedding

Jeremie Tafforeau[(1)], *Thierry Artieres*[(1)(2)], *Benoit Favre*[(1)], *Frederic Bechet*[(1)]

(1) Aix Marseille Universite, CNRS-LIF
(2) Ecole Centrale Marseille
firsname.lastname@lif.univ-mrs.fr

Abstract

Word embeddings have become ubiquitous in NLP, especially when using neural networks. One of the assumptions of such representations is that words with similar properties have similar representation, allowing for better generalization from subsequent models. In the standard setting, two kinds of training corpora are used: a very large unlabeled corpus for learning the word embedding representations; and an in-domain training corpus with gold labels for training classifiers on the target NLP task. Because of the amount of data required to learn embeddings, they are trained on large corpus of written text. This can be an issue when dealing with non-canonical language, such as spontaneous speech: embeddings have to be adapted to fit the particularities of spoken transcriptions. However the adaptation corpus available for a given speech application can be limited, resulting in a high number of words from the embedding space not occurring in the adaptation space. We present in this paper a method for adapting an embedding space trained on written text to a spoken corpus of limited size. In particular we deal with words from the embedding space not occurring in the adaptation data. We report experiments done on a Part-Of-Speech task on spontaneous speech transcriptions collected in a call-centre. We show that our word embedding adaptation approach outperforms state-of-the-art Conditional Random Field approach when little in-domain adaptation data is available.

Index Terms: word embeddings, deep neural network, spontaneous speech, POS tagging.

1. Introduction

Representation learning has emerged as a key issue in machine learning, and has led to major breakthroughs in computer vision and natural language processing [1, 2]. In particular researchers in NLP have focused on learning dense low dimensional (hundreds) representation space of words [3, 4], called embeddings, which model both semantic and syntactic information.

The benefits of such representations are (1) that they offer a lower computational complexity when used as input of classifiers such as neural networks, and (2) that words with similar properties have similar representations, allowing for better generalisation from subsequent models, e.g. for words not covered by targeted task training data. This strategy has been applied successfully for many classical NLP tasks such as information retrieval, language modelling [5], machine translation [6], as part-of-speech tagging, named entity recognition [7], syntactic parsing [8, 9], semantic role labeling [10], etc.

In the standard setting of embedding space usage, two kinds of training corpora are used: a very large unlabelled corpus (C_{emb} for *embedding corpus*) on which word representations are learned, and a smaller in-domain training corpus with gold labels for training classifiers on the target NLP task (C_{task}). It is assumed that the syntactic/semantic contexts learned in C_{emb} are coherent with those of the in-domain corpus, and since C_{emb} has a much wider coverage than C_{task}, therefore all the words of C_{task} should have a representation in C_{emb}.

When the adaptation corpus represents a different register of language than the standard canonical written language (e.g. Wikipedia) covered by C_{emb}, these assumptions are not necessarily true. This is the case when embeddings are used to process spontaneous speech transcriptions of a specific domain for which few manual transcriptions are available. This situation is rather usual when processing call-centre data considering the difficulties of collecting and transcribing human-human conversations for each use-case.

If no adaptation is performed, embeddings learned on written text might be too far from word contexts found in spontaneous speech. For instance, words like “yes” or “no” are seldom used in written text while they a very frequent in dialogs. Other words, such as “like” change their most common use between text (preposition, noun, verb) and speech (adverb). Even by adapting embeddings on the C_{task} corpus, because of its small size, only a small portion of the embedding space will be affected.

This paper presents a method that addresses these issues by both adapting an embedding space thanks to a small adaptation corpus, for a specific task, then by generalizing this adaptation to all words of the original embedding space, in particular to those not occurring in the adaptation corpus. Our contributions are as follows:

- Integration of word embeddings in a neural network performing a target task (here, part-of-speech tagging)
- Embedding adaptation for words of the target task corpus through *refinement* (initialization of a hidden layer with original embeddings before training the neural network)
- *Artificial refinement* for those out-of-vocabulary (OOV) words unseen in the target task training data.

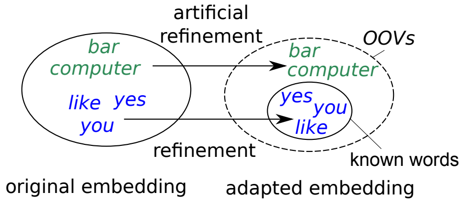

Figure 1: *Illustration of the proposed adaptation process*

The applicative framework of this paper is the lightly supervised adaptation of a POS tagger to process spontaneous speech transcriptions with very small amount of annotated training data and a very large unlabelled text corpus with a language-type mismatch (written text *v.s.* spontaneous speech). The proposed approach is illustrated in Figure 1.

Processing spontaneous speech is challenging: disfluencies and oral-specific syntactic constructs lead to a drop in performance when using models trained for written text. Although the task targeted is Part-Of-Speech tagger, the strategies proposed are task-independent and don't make any assumptions of the kinds of label to predict.

We show that our adaptation strategy improves over a state-of-the-art baseline using a CRF tagger when small amount of data is available to train the models, and when there is a mismatch between C_{emb} and the target corpus.

2. Related work

OOV word handling in NLP tasks is dependent on the feature space used to encode data. Features can be computed from the sequence of characters composing the word (e.g. morphological, suffix and prefix features [11, 12]) in order to steer the classifier's decision when the form is unknown. Contextual features try to take advantage of the words in the vicinity of the OOV, such as n-grams in sequence models; contexts can be gathered in external corpora or using web queries [13]. OOVs can also be replaced by surrogates which have the same distributional properties, such as word clusters which have proved to be effective in many tasks [14].

Relaying on an embedding space for encoding words opens new possibilities for OOV handling: the availability of large corpora for learning embeddings and methods to process them [3] reduces the number of OOVs. For words unknown from the task training corpus (C_{task}) but occurring in the embedding corpus (C_{emb}), a similarity distance in the embedding space can be used to retrieve the closest known words and use its characteristics.

For words not in C_{emb}, a generic OOV model is used. These methods are reviewed and evaluated in [9] on a dependency parsing task showing that a small performance gain can be obtained when little training data is available. We propose in this paper to push forward these experiments by extending the embedding space for OOVs not in C_{task}.

3. Embeddings adaptation for POS tagging of spontaneous speech

Our work aims at learning a Neural Network for POS tagging where the input layer is a lookup layer (also called embedding), that we note Φ, which transforms a sequence of words $(w_1, ..., w_T)$ to a sequence of low dimensional vectors $(\Phi(w_1), ..., \Phi(w_T))$. The transformation Φ is initialized as the embedding learnt using the approach in [3], noted Φ_0. It is further refined during the supervised training of the neural net on the POS tagging task. We note Φ_r the (refined) final embedding.

The neural net architecture we use is similar to [10] and is illustrated in Figure 2. It is a two hidden layers net whose input is a window of 5 successive words in a sentence and morphological features about the word of interest. As morphological features, we consider three boolean values about capitalisation, number and non alpha-numeric characters presence. In addition, we also use a word representation based on a bag of character bi-grams. For example, the word "boat" is represented as $\langle$bo, ba, bt, oa, ot, at$\rangle$. In order to restreint the dimension of this representation, we only consider most frequent bi-grams, covering 90% of occurrences.

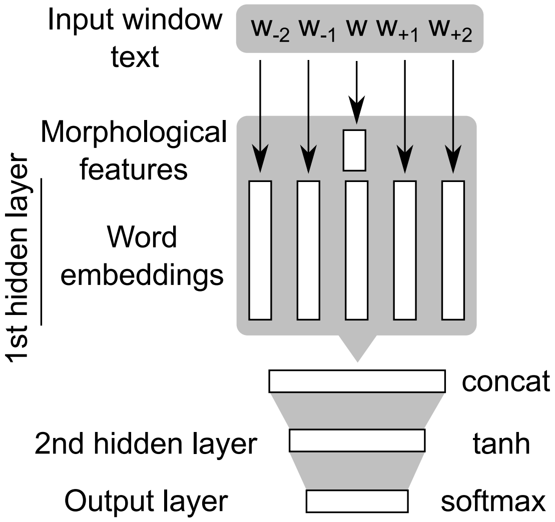

Figure 2: *Our system is a neural network which takes as input a window of words centered on the word to label and morphological features associated to the word of interest. It is learnt to predict the part-of-speech tag of the word at the middle of the input window.*

The first layer is a *lookup* layer that replaces each of these 5 words by their embedding. It is implemented as a concatenation of 5 parallel hidden layers of size 200, which is the dimension of the embedding space. Each of these hidden layers has a huge input layer whose size equals the size of the vocabulary, they share the same set of weights. The input vector corresponding to a particular word is a sparse vector, with a 1 at the word position number and null otherwise. For a given input windows of 5 words the hidden layer is then a vector of size 5×200. The embedding of the i^{th} word of the vocabulary is then encoded in the weights from the i^{th} input neuron to the hidden layer. This first hidden layer is fully connected to a second nonlinear hidden layer (in our experiments it has 256 neurons) which is itself fully connected to an output layer with 20 neurons (one neuron per par-of-speech tag, i.e. per class). We note $\theta = (W, E)$ the set of parameters of the whole model, where E is the embedding space while W corresponds to every other neural network layers weights.

We use softmax outputs which means that the actual outputs of this neural net for a given input x, $f_\theta(x) = ([f_\theta(x)]_i)_{i=1..N}$ (where N stands for the number of classes) and $[f_\theta(x)]_i$ is the i^{th} component of the output vector $f_\theta(x)$ corresponding to the score for the i^{th} class, are normalized to sum to 1 and may be interpreted as probabilities $p(i|x, W, E)$. The neural net is trained using a training set $\mathcal{T}$ of pairs (x, y) of (window of words, POS tag) to maximize a log-likelihood criterion:

$$\hat{\theta} = \underset{(W,E)}{argmax} \sum_{(x,y)\in\mathcal{T}} \log\ p(y|x, W, E)$$

where x corresponds to a word window and its associated morphological features, and y represents the corresponding tag. Optimization is performed through stochastic gradient descent (SGD), i.e. by iteratively selecting a random sample (x, y) and making a gradient step :

$$\theta \leftarrow \theta \,+\, \lambda\, \frac{\partial\, log\, p(y|x, W, E)}{\partial \theta},$$

where λ is the learning rate. In this way, backpropagating the gradient through the whole network yield refined embeddings which become more and more task/domain-specific all along the training, from Φ_0 to Φ_r. This may be especially interesting when the corpus used for learning embedding is different from the one used to learn the task-specific model (POS tagging in our case).

4. Artificial refinement of an OOV embedding

The embedding refinement method proposed in the previous section can only be applied to words belonging to the adaptation corpus. All the other words remain unchanged. This is often the case when dealing with supervised learning since annotated corpora are expensive to produce and are therefore much smaller than those used for learning word embeddings as in [3].

This yields to an heterogeneous embedding space where some word embeddings have been tuned while others have not. Let note V_{emb}, V_{task} and V_{test} the vocabulary used for learning embeddings, for learning the POS tagger and the vocabulary one may encounter in tests. Because the embedding corpus is usually very large we will be very likely to have $V_{\text{task}} \subseteq V_{\text{emb}}$, $V_{\text{test}} \subseteq V_{\text{emb}}$. However at the same time many words may occur in test while not being in the training corpus, as illustrated by figure 1.

We propose here to artificially refine these OOV embeddings. In order to do so, we simulate the refinement of a word t embedding, $\Phi_0(t)$, thanks to its K nearest neighbours in the original embedding space. We make the hypothesis that if word t occurs in the POS tagging training set C_{task}, it would have been refined in a similar way than its closest words in the original embedding space.

Let note $(n_k)_{k=1..K}$ the K words from $V_{\text{emb}} \cap V_{\text{task}}$ whose initial embeddings $\Phi_0(n_k)$ are the closest to $\Phi_0(t)$. These word embeddings have been shifted from $(\Phi_r(n_k) - \Phi_0(n_k))$. We propose to compute an artifical refinement $\Phi_r(t)$ as a weighted sum of the shifts applied to its nearest neighbours.

$$\Phi_r(t) = \Phi_0(t) + \sum_{k=1}^{K} \alpha_k (\Phi_r(n_k) - \Phi_0(n_k))$$

where the mixing coefficients α are positive real values that sum to 1. In our experiments we define these coefficients as being proportional to the cosine similarity s between t and n_k:

$$s(t, n_k) = \frac{\Phi_0(t) \cdot \Phi_0(n_k)}{|\Phi_0(t)| \times |\Phi_0(n_k)|}$$

5. Experiments

The two datasets used in our experiments are the French RATP-DECODA corpus (546K words) for the in-domain labelled corpus and the French part of Wikipedia for the unlabelled C_{emb} corpus (357M words). The RATP-DECODA corpus [15] collected within the DECODA project is made of 1514 conversations over the phone recorded at the Paris public transport call center. We used the same train/test partition as described in [16].

The train section C_{task} contains 521K words and the test section C_{test} 25K words. In order to test our adaptation strategy with different sizes of adaptation corpus, we split C_{task} into 5 sections: D_{10} contains only the first 10 dialogs (in chronological order) of C_{task}, D_{50} the first 50 dialogs and D_{100} the first 100 dialogs. Finally D_{all} contains the whole training corpus.

For each partition we have 4 kinds of words in the test corpus, described in Table 1.

	V_{task}	V_{emb}	V_{test}
W1	X	X	X
W2		X	X
W3	X		X
W4			X

Table 1: *Categorization of words in* V_{test}

W4 are the *full* Out-Of-Vocabulary words, only occurring in the test corpus; **W3** and **W2** are *partial* OOVs since they belongs either to V_{task} or V_{emb}. The amount of **W2** and **W4** words decreases when the amount of training data increases. The adaptation strategy presented in this study is focusing on **W1** and **W2** words: **W1** words are adapted according to the method presented in section 3; **W2** words are adapted with the artifical refinement method of section 4. Table 2 presents the distributions of words for the 4 training corpus sections.

C_{task}	$\lvert C_{\text{task}} \rvert$	**W1**	**W2**	**W3**	**W4**
D_{10}	1,662	18,323	5,025	709	1,336
D_{50}	10,833	21,189	2,159	1,131	914
D_{100}	23,251	21,808	1,540	1,285	760
D_{all}	521,377	23,083	265	1,562	483

Table 2: *Distribution of words in the test corpus* C_{test} *according to the different training partitions*

Our experimental results are presented in Table 3 and in Table 4. Four systems are compared:

- **CRF** is a state-of-the-art Conditional Random Field tagger using lexical context as well as morphological features for predicting the best sequence of POS labels while handling OOVs. This tagger is part of the MACAON NLP tool suite [17] and has already been used on spoken data in [16]. This tagger is used as a baseline system.
- **NN** corresponds to our neural network without the embedding refinement process. In this system, words embeddings are used as input but are not part of trainable parameters (only W weights are trained).
- **NN+ER** integrates the word embeddings refinement method proposed in section 3. OOV embeddings are kept to their original values.
- **NN+ER+AER** is our full system, where known words embeddings are naturally refined through the neural net training while OOV embeddings are artificial refined as described in previous section.

As we can see, our OOV handling strategy systematically outperforms the CRF standard one. The gain is particularly significant when very small amount of training data is available (D10 and D50), but even when the full training corpus is used, we observed improvements.

Train	Model	**All**	**W1**	**W2**
D_{10}	CRF	12.33	5.6	32.6
	NN+ER+AER	10.28	5.1	21
D_{50}	CRF	5.81	4.2	18
	NN+ER+AER	5.62	3.8	11.2
D_{100}	CRF	4.74	3.8	15.2
	NN+ER+AER	4.81	3.5	12
D_{all}	CRF	3.18	3	10.9
	NN+ER+AER	3.27	3	8.7

Table 3: *POS error rate (in %) computed on all the test set (**All**) or restricted to **W1** and **W2** categories as a function of the training corpus size. State-of-art baseline CRF tagger v.s. our proposed neural network model*

Train	Model	**All**	**W1**	**W2**
D_{10}	NN	13.01	5.7	34.5
	NN+ER	10.55	5.1	22.5
	NN+ER+AER	10.28	5.1	21
D_{50}	NN	6.25	4.1	17.5
	NN+ER	5.78	3.8	12.4
	NN+ER+AER	5.62	3.8	11.2
D_{100}	NN	5.08	3.7	15.8
	NN+ER	4.89	3.5	12.9
	NN+ER+AER	4.81	3.5	12
D_{all}	NN	3.32	3.1	10.6
	NN+ER	3.29	3	9.8
	NN+ER+AER	3.27	3	8.7

Table 4: *POS error rate (in %) as a function of the training corpus size and of the learning strategy. **NN** is a neural network with constant embeddings unlike **NN+ER** where embedding are refined over time but without OOV handling strategy. **NN+ER+AER** correspond to our full system with embedding refinement and OOV embedding artificial refinement.*

However, as the proportion of OOV in the test corpus decreases when the size of the training corpus increases, the overall gain is modest. Furthermore, we did not present results about OOV words without embeddings (categories **W3** and **W4**) because it is orthogonal to our approach but it probably explains the CRF slightly better overall performance w.r.t. our system. An interesting line of future work will be to take these kind of OOVs into account with a specific process in order to close this gap.

Contrastive results are given in table 4. We check the improvement obtained with each refinement method. The embedding refinement (**ER**) method leads to significant overall improvements when considering small training data (D10 and D50). The artificial embedding refinement (**AER**) method provides a constant gain over OOV recognition performance, even when the whole training corpus is used. These results validate our adaptation approach.

6. Conclusion

Processing spontaneous speech is challenging: disfluencies and oral-specific syntactic constructs lead to a drop in performance when using models trained for written text. We show in this study that our OOV processing strategies can help when a small amount of data is available to train the models and/or when there is a mismatch in the language style between the corpus used to learn the embeddings and the target corpus.

An interesting line of future work will be to add tag sequence global optimisation, using *Recurrent Neural Network* architectures and *Long-Short Term Memory* neurons, instead of considering independently each word in a sequence. Moreover, we plan to investigate the relevance of our approach on larger corpora and on more difficult NLP tasks such as syntactic parsing and semantic role labelling.

7. Acknowledgements

The research leading to these results has received funding from the European Union - Seventh Framework Programme (FP7/2007-2013) under grant agreement n 610916 SENSEI.

8. References

[1] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” vol. abs/1207.0580, 2012.

[2] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, “Reasoning With Neural Tensor Networks For Knowledge Base Completion,” in *Advances in Neural Information Processing Systems 26*, 2013.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” vol. abs/1301.3781, 2013.

[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Advances in Neural Information Processing Systems 26*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2787–2795.

[5] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, “A neural probabilistic language model,” *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[6] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[7] B. Tang, H. Cao, X. Wang, Q. Chen, and H. Xu, “Evaluating word representation features in biomedical named entity recognition tasks,” *BioMed research international*, vol. 2014, 2014.

[8] M. Bansal, K. Gimpel, and K. Livescu, “Tailoring continuous word representations for dependency parsing,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014.

[9] J. Andreas and D. Klein, “How much do word embeddings encode about syntax,” in *Proceedings of ACL*, 2014.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” in *the Journal of Machine Learning Research 12*, 2011, pp. 2461–2505.

[11] T. Schnabel and H. Schütze, “FLORS: Fast and Simple Domain Adaptation for Part-of-Speech Tagging,” vol. 2, February 2014, pp. 15–26.

[12] C. D. Santos and B. Zadrozny, “Learning character-level representations for part-of-speech tagging,” in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, T. Jebara and E. P. Xing, Eds. JMLR Workshop and Conference Proceedings, 2014, pp. 1818–1826.

[13] S. Umansky-Pesin, R. Reichart, and A. Rappoport, “A multi-domain web-based algorithm for pos tagging of unknown words,” in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING ’10. Association for Computational Linguistics, 2010, pp. 1274–1282.

[14] O. Owoputi, B. O’Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith, “Improved part-of-speech tagging for online conversational text with word clusters.” in *HLT-NAACL*, 2013, pp. 380–390.

[15] F. Bechet, B. Maza, N. Bigouroux, T. Bazillon, M. El-Beze, R. D. Mori, and E. Arbillot, “Decoda: a call-centre human-human spoken conversation corpus,” in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12)*. European Language Resources Association (ELRA), may 2012.

[16] A. Nasr, F. Bechet, B. Favre, T. Bazillon, J. Deulofeu, and A. Valli, “Automatically enriching spoken corpora with syntactic information for linguistic studies,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, 2014, pp. 854–858.

[17] A. Nasr, F. Béchet, J. Rey, B. Favre, and J. Le Roux, “Macaon: An nlp tool suite for processing word lattices,” *Proceedings of the ACL 2011 System Demonstration*, pp. 86–91, 2011.