Phrase Representations for Multiword Expressions

Joël Legrand[1,2] and **Ronan Collobert***[3,1]
[1] Idiap Research Institute, Martigny, Switzerland
[2] Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[3] Facebook AI Research, Menlo Park (CA), USA

Abstract

Recent works in Natural Language Processing (NLP) using neural networks have focused on learning dense word representations to perform classification tasks. When dealing with phrase prediction problems, is is common practice to use special tagging schemes to identify segments boundaries. This allows these tasks to be expressed as common word tagging problems. In this paper, we propose to learn fixed-size representations for arbitrarily sized chunks. We introduce a model that takes advantage of such representations to perform phrase tagging by directly identifying and classifying phrases. We evaluate our approach on the task of multiword expression (MWE) tagging and show that our model outperforms the state-of-the-art model for this task.

1 Introduction

Traditional NLP tasks such as part-of-speech (POS) tagging or semantic role labeling (SRL) consists in tagging each word in a sentence with a tag. Another class of problems such as Named Entity Recognition (NER) or shallow parsing (chunking) consists in identifying and labeling phrases (*i.e.* groups of words) with predefined tags. Such tasks can be expressed as word classification problems by identifying the phrase boundaries instead of directly identifying the whole phrases. In practice, this consists in prefixing every tag with an extra-label indicating the position of the word inside a phrase (at the beginning (B), inside (I), at the end (E), single word (S) or not in a phrase (O)). Different schemes have been used in the literature, such as the IOB2, IOE1 and IOE2 schemes (Sang and Veenstra, 1999) or IOBES scheme (Uchimoto et al., 2000) with no clear predominance.

These tasks have been tackled using various machine learning methods such as Support Vector Machines (SVM) for POS tagging (Giménez and Màrquez, 2004) or chunking (Kudoh and Matsumoto, 2000), second order random fields for chunking (Sun et al., 2008) or a combination of different classifiers for NER (Radu et al., 2003). All these approaches use carefully selected handcrafted features.

Recent studies in NLP introduced neural network based systems that can be trained in an end-to-end manner, using minimal prior knowledge. These models take advantage of continuous representations of words. In Collobert et al. (2011) the authors proposed a deep neural network, which learns the word representations (the features) and produces IOBES-prefixed tags discriminatively trained in an end-to-end manner. This system is trained using a conditional random field (Lafferty et al., 2001) that accounts for the structure of the sentence. This architecture has been applied to various NLP tasks, such as POS tagging, NER or semantic role labeling and achieves state-of-the-art performance in all of them.

In this paper, we propose to learn fixed-size continuous representations of arbitrarily sized chunks by composing word embeddings. These representations are used to directly classify phrases without using the classical IOB(ES) prefixing step. The proposed approach is evaluated on the task of multiword expression (MWE) tagging. Using the SPRML 2014 data for French MWE tagging (Seddah et al., 2013), we show that our phrase representations are able to capture enough knowledge to perform on par with the IOBES-based model of Collobert et al. (2011) applied to MWE

*All research was conducted at the Idiap Research Institute, before Ronan Collobert joined Facebook AI Research tagging. Furthermore, we show that our system outperforms the winner of the SPMRL (Syntactic Parsing of Morphologicaly Rich Language) 2013 shared task for MWE tagging (Constant et al., 2013) which is currently the best published system.

2 The model

The proposed model computes fixed-size continuous vectors of arbitrarily sized chunks which are then used as inputs to a classifier. Every possible window of sizes from 1 to K (K being the maximum size) is projected onto a common vector space (the same for all k), using a different neural network for each size k. The resulting representations are passed on to a classifier which outputs a score for every possible tag. To ensure that a word belongs to one chunk at most, decoding is performed using structured graph decoding using the Viterbi algorithm.

2.1 Word representation

Given an input sentence $S = w_1, \ldots, w_N$, each word is embedded into a D-dimensional vector space by applying a lookup-table operation (Bengio et al., 2000):

$$LT_W(w_n) = W_{w_n}$$

where the matrix $W \in \mathbb{R}^{D \times |\mathcal{W}|}$ represents the parameters of the lookup layer. Each column $W_n \in \mathbb{R}^D$ corresponds to the vector embedding of the n^{th} word in the dictionary $\mathcal{W}$.

Additional features, such as part-of-speech tags, can be used by using a different lookup table for each discrete feature. The input becomes the concatenation of the outputs of all these lookup-tables. For simplicity, we consider only one lookup-table in the rest of the architecture description.

2.2 Phrase representation

We denote k-window a window of size $k \in [1, K]$ where K is the maximum window size. Phrase representations for all k-windows within a given sentence are produced by looking, for all sizes from 1 to K, at all successive windows of text, sliding over the sentence, from position 1 to $N - K + 1$. Formally, if we denote

$$\begin{aligned} x_{n,k} =& [LT_W(w_{n-c}), \ldots, LT_W(w_n) \\ & , \ldots, \\ & , LT_W(w_{n+k-1}), \ldots, LT_W(w_{n+k-1+c})] \end{aligned}$$

the concatenated word representations corresponding to the n^{th} k-window (c being the context from each side of the the k-window), its representation is given by

$$r_{n,k} = M_k^1 x_{n,k},$$

where $M_k^1 \in \mathbb{R}^{(k+2c)D \times nhu}$ is a matrix of parameters and nhu the dimension of the phrase representations (which is the same for all k). Words outside the sentence boundaries are assigned a special "PADDING" embedding.

2.3 Phrase scoring

We denote $\mathcal{T}$ the set of tags and $\mathcal{T}_k$ the set of tags for a k-window. We denote $t_k \in \mathcal{T}_k$ the tag $t \in \mathcal{T}$ for a k-window. The scores for all k-windows are computed by a linear layer, using their corresponding representations as input. Formally, the score for the n^{th} k-window are given by

$$s_{n,k} = tanh(M^2 r_{n,k}),$$

where $M^2 \in \mathbb{R}^{nhu \times |\mathcal{T}|}$ is a matrix of parameters. We define s_{n,t_k} the score for the tag $t_k \in \mathcal{T}_k$ starting at the position $n < N - k + 1$.

2.4 Structure tag inference

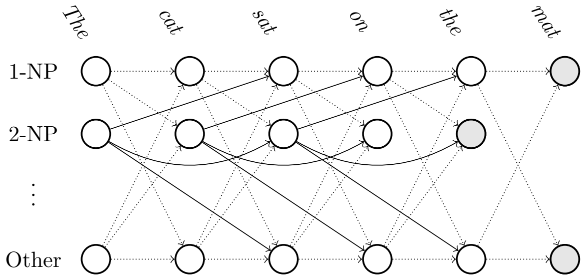

Figure 1: Constrained graph for structured inference. Each node is assigned a score from the scoring layer. For instance, the first node of the line 2-NP correspond to the score for the tag NP for the phrase "the cat". Nodes in gray represent final nodes.

The scoring layer outputs a matrix of $|\mathcal{T}_k| \times (N - k + 1)$ scores for each window size $k \in K$.

The next module (see Figure 1) of our system is a structured graph G constrained in order to ensure that a word is tagged only once. Each node G_{n,t_k} is assigned the score s_{n,t_k} (the score of the tag $t_k \in \mathcal{T}_k$ starting at the position $n < N-k+1$) from the scoring layer. Only transitions from node G_{n,t_k} to node $G_{n+k,t'_{k'}}$ (with $n + k <= N$) are possible since a word cannot be tagged twice along the same path. The Viterbi algorithm is an efficient choice to find the best path in the lattice. The score for a sentence S of size N along a path of tags $[t]_1^{N_t}$ is then given by the sum of the tag scores:

$$s(S, [t]_1^{N_t}, \theta) = \sum_{n=1}^{N_t} s_{n,t_k}$$

where θ represents all the trainable parameter.

2.5 Training

The proposed neural network is trained by maximizing the likelihood over the training data, using stochastic gradient ascent. Following Collobert et al. (2011), the score $s(S, [t]_1^{N_t}, \theta)$ can be interpreted as a conditional probability by exponentiating this score and normalizing it with respect to all possible path scores. Taking the log, the conditional probability of the true path $[t]_1^{N_t}$ is given by

$$\begin{aligned}\log p(s(S, [t]_1^{N_t}, \theta)) &= s(S, [t]_1^{N_t}, \theta)\\ &\quad - \log(\sum_u s(S, [u]_1^{N_u}), \theta)\end{aligned}$$

Following Rabiner (1990), the normalization term (second term of this equation) can be computed in linear time thanks to a recursion similar to the Viterbi algorithm. The whole architecture (including the input feature, phrase representations and scoring layer) is trained through the graph in order to encourage valid paths of tags during training, while discouraging all other paths.

3 Experiments

3.1 Multiword expression

Multiword expressions are groups of tokens which act as single units at some level of linguistic analysis. They cover a wide range of linguistic constructions such as idioms (”kick the bucket”), noun compound (”traffic light”) or fixed phrases (”ad hoc”). As they can carry meaning that can not be inferred directly from the meaning of individual constituents (as for idioms), they are difficult to handle by automatic systems and represent a key issue for many NLP systems addressing, for instance, machine translation and text generation tasks.

3.2 Corpus

Experiments were conducted on the SPMRL french corpus provided for the Shared Task 2013 (Seddah et al., 2013). This dataset provides 14.7k sentences (443k tokens) with 22.6k identified MWE. A given MWE is defined as a continuous sequence of terminals, plus a POS tag among the 10 possible POS tags. As presented in Table 1, a wide majority of the chunks are 2-chunks or 3-chunks (91.2%).

Chunk size	2	3	4	5	5+
#chunk	11108	10188	1702	309	250
percentage	47.2	43.2	7.2	1.3	1.1

Table 1: Number of k-sized chunks in the training corpus

3.3 Evaluation

We evaluate the performance of the proposed network on MWE tagging using the three metrics described in Seddah et al. (2013), reporting for each of them the recall, precision and F-score. MWE correspond to the full MWEs, in which a predicted MWE counts as correct if it has the correct span (same group as in the gold data). MWE+POS is defined in the same fashion, except that the predicted MWE counts as correct if it has both correct span and correct POS tag. COMP correspond to the non-head components of MWEs: a non-head component of MWE counts as correct if it is attached to the head of the MWE, with the specific label indicating that it is part of an MWE.

3.4 Baseline models

We compare the proposed model to our implementation of the IOBES-based model described in Collobert et al. (2011), applied to MWE tagging. We also report the results of the LIGM-Alpage architecture which obtained the best results for French SPMRL 2013 MWE recognition shared task (Constant et al., 2013). Their system is based on Conditional Random Fields (CRF) (Lafferty et al., 2001) and on external lexicons which are known to greatly improve MWE segmentation (Constant and Tellier, 2012).

3.5 Setup

The network is trained using stochastic gradient descent over the training data, until convergence on the validation set. Hyper-parameters are tuned on the validation set. The look-up table size for the words is 64. Word embeddings are pre-trained by performing PCA on the matrix of word co-occurrences (Lebret and Collobert, 2014) using Wikipedia data. These embeddings are fine-tuned during the training process. As additional features, we only use the part-of-speech tags obtained using the freely available tool MarMoT (Mueller et al., 2013)[1]. The POS-tag embedding size is 32. The context size is $c = 2$ The maximum size for a window is $K = 7$. The common embedding size for the k-window is $nhu = 300$. We fix the learning rate to 0.01. Following Legrand and Collobert (2015), to prevent units from co-adapting, we adopt a dropout regularization strategy (Srivastava et al., 2014) after every lookup-table, as the capacity of our network mainly lies on the input embeddings.

For the IOBES-based model, we use the following parameters: the context size is set to 2, word and tags feature sizes are 64 and 32 respectively, the hidden layer size is 300 and the learning rate is 0.001. We use the same dropout regularization strategy and the same word initialization as for the proposed model.

4 Results

We first compare our approach with the IOBES-model from Collobert et al. (2011). Table 2 presents the results obtained for the two models. We see that, our model performs on par with the IOBES-based model. Interestingly, adding the POS features has little effect on the performance for MWE identification but helps to determine the MWE POS-tags.

In Table 3, we compare our model with the winner of the SPMRL 2013 shared task for MWE recognition (Constant et al., 2013). Both the IOBES and chunk based models are obtained using an ensemble of 5 model and averaging the obtained scores. We see that both our model and the IOBES-based model outperform this state-of-the-art model.

	COMP	MWE	MWE+POS
IOBES-model	79.4	78.5	75.4
+ WI	80.8	80.1	76.7
+ WI + POS	80.8	80.1	77.6
Chunk-model	79.1	78.3	75.2
+ WI	80.7	79.6	76.4
+ WI + POS	80.9	79.8	77.5

Table 2: Results on the test corpus (4043 MWEs) in terms of F-measure. WI stands for word initialization.

	COMP	MWE	MWE+POS
LIGM-Alpage	81.3	80.7	77.5
IOBES-model	81.4	80.7	78.2
Chunk-model	81.3	80.7	78.1

Table 3: Results on the test corpus (4043 MWEs) in terms of F-measure.

5 Representation analysis

As the proposed chunk-based model produces continuous phrase representations, it allows for phrase comparison. Table 4 presents some of the closest neighbors (in terms of Euclidean distance) for some chosen phrases. We see that close representations correspond to semantically close phrases.

président de la république
chef de l'état
présidence de la république
ministre de l'intérieur

évasion fiscale
fraude fiscale
détournements financiers
libéralisme sauvage

impôt sur le revenu
impôt sur la fortune
impôt sur le patrimoine
impôts sur la fortune

Table 4: Closest neighbors for three input phrases in terms of euclidean distance.

[1] The tags used are available here: http://cistern.cis.lmu.de/marmot/models/CURRENT/

6 Conclusion

In this paper, we proposed a neural network model that learns fixed-size continuous representations of arbitrarily-sized chunks by composing word embeddings. These representations are used to directly identify and classify phrases. Evaluating our model on the task of multiword expression tagging, we showed that the proposed representations perform on par with a baseline IOBES-based system. We also showed that it outperforms the model obtaining the best published performance for this task while not using any external lexicon and relying on few input features. As the proposed model computes phrase representations, it allows for comparison between phrases. In the future, the potential of this approach for higher-level tasks such as bilingual word alignment are to be explored.

References

Y. Bengio, R. Ducharme, and P. Vincent. A Neural Probabilistic Language Model. In *NIPS*, 2000.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2011.

M. Constant, M. Candito, and D. Seddah. The LIGM-Alpage Architecture for the SPMRL 2013 Shared Task: Multiword Expression Analysis and Dependency Parsing. In *Fourth Workshop on Statistical Parsing of Morphologically Rich Languages*, 2013.

J. Giménez and L. Màrquez. Svmtool: A general pos tagger generator based on support vector machines. In *LREC*, 2004.

T. Kudoh and Y. Matsumoto. Use of support vector learning for chunk identification. In *Proceedings of the 2Nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7*, 2000.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Int. Conf. on Machine Learning (ICML 2001)*, 2001.

R. Lebret and R. Collobert. Word Embeddings through Hellinger PCA. In *Proc. of EACL*, 2014.

J. Legrand and R. Collobert. Joint RNN-Based Greedy Parsing and Word Composition. In *Proceedings of ICLR*, 2015.

T. Mueller, H. Schmid, and H. Schütze. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

L. R. Rabiner. Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. 1990.

F. Radu, I. Abe, J. Hongyan, and Z. Tong. Named entity recognition through classifier combination. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, 2003.

E. F. T. K. Sang and J. Veenstra. Representing text chunks. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, 1999.

D. Seddah, R. Tsarfaty, S. Kübler, M. Candito, J. D. Choi, R. Farkas, J. Foster, I. Goenaga, K. Gojenola, Y. Goldberg, S. Green, N. Habash, M. Kuhlmann, W. Maier, J. Nivre, A. Przepiórkowski, R. Roth, W. Seeker, Y. Versley, V. Vincze, M. Wolińsk, A. Wróblewska, and E. Villemonte De La Clergerie. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, 2013.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 2014.

X. Sun, L. Morency, D. Okanohara, and J. Tsujii. Modeling latent-dynamic in shallow parsing: A latent conditional model with improved inference. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, 2008.

K. Uchimoto, Q. Ma, M. Murata, H. Ozaku, and H. Isahara. Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000.