Multi-label, Multi-class Classification using Polylingual Embeddings

Georgios Balikas and Massih-Reza Amini

[1] University of Grenoble-Alpes/Coffreo
georgios.balikas@imag.fr/gbalikas@coffreo.com
[2] University of Grenoble-Alpes
massih-reza.amini@imag.fr

Abstract. We propose a Polylingual text Embedding (PE) strategy, that learns a language independent representation of texts using Neural Networks. We study the effects of bilingual representation learning for text classification and we empirically show that the learned representations achieve better classification performance compared to traditional bag-of-words and other monolingual distributed representations. The performance gains are more significant in the interesting case where only few labeled examples are available for training the classifiers.

1 Introduction and Preliminaries

In this work we propose a mechanism for combining distributed representations of documents in different languages. Each document in a given language is first translated using an existing Machine Translation (MT) tool. The rationale behind is that translation offers the possibility to enrich and disambiguate the text, especially for short documents. Documents are then represented by aggregating the embeddings of their associated text spans in each language [7,9] using a non-linear auto-encoder (AE). The AE is trained on their concatenated representations and a classifier is finally trained in the polylingual space outputed by the auto-encoder. Our classification results in a subset of the publicly available Wikipedia show that our approach yields improved classification performance compared to the case where a classical bag-of-words space is used for document representation, especially in the case where the size of the training set is small.

Neural Networks have recently shown promising results in several machine learning and information extraction tasks [12,13,2]. For text classification, the use of embeddings as inputs or initializations to more complex architectures has been investigated and, for example, [4,5] study the benefits of embeddings of sentence-length spans (sentences and/or questions). In the multilingual setting, [3] proposed an approach to learn bilingual embeddings exploiting parallel and non-parallel text in the languages, [1] proposed to use correlated components analysis, together with small bi-lingual lexicons, to learn how to project embeddings in two separate languages into a common representation space and [6] proposed an approach similar to ours that uses an auto-encoder to learn bilingual representations.

In the next section we present our polylingual embedding strategy. In the experimental part (Section 3), we empirically show that the learned representations constitute better classification features compared to several baselines and their value can strongly benefit classification settings with few labeled examples. We discuss these results in Section 4 and conclude in Section 5.

2 The proposed approach

Monolingual distributed representations (DRs) project text spans into a language-dependent semantic space where spans with similar semantics are closer in that space. Here, we aim to combine two distributed representations of documents corresponding to the original document and its translation using an auto-encoder. We will refer to those combined representations as *Polylingual Embeddings* (PE). We suppose that the auto-encoder will disentangle the language-dependent factors and will learn robust representations on its hidden layer encoding as illustrated in Figure 1. Given a document d_i in English, we first translate it into French using a commercial translator, we then generate the distributed representations of the document and its translation $\{\mathcal{G}^\ell(d_i)\}_{\ell=1}^2$, and then aggregate those DRs using an auto-encoder (Algorithm 1).

The auto-encoder is learned over all concatenated distributed representations of documents using a stochastic back-propagation algorithm. In this work we consider two strategies to create the DR of each document. The first one is based on average pooling, where word representations are first obtained using the word2vec tool [8]. DR of documents, i.e. functions $(\mathcal{G}^\ell)_{\ell\in\{1,2\}}$, are then obtained by averaging the vectors of words contained in them. In this study we consider the *continuous bag of words* (cbow) and the *skip-gram* models that generate word representations. The second strategy is based on the *distributed Memory Model of paragraph vectors* (DMMpv) and *distributed bag-of-words of paragraph vectors* (DBOWpv) models [7], that extend cbow and skip-gram respectively. In this case, $(\mathcal{G}^\ell)_{\ell\in\{1,2\}}$ are defined by the output of the models without further processing.

Require: $\{\mathcal{G}^\ell(d_i)\}_{\ell=1}^2$, a trained AE
1: **for** each document d_i **do**
2: Concatenate $\mathcal{G}^1(d_i)$ and $\mathcal{G}^2(d_i)$
3: Get PE representation of d_i as the hidden encoding of the AE fed with the concatenation
4: **end for**

Algorithm 1: The process of generating PE representations

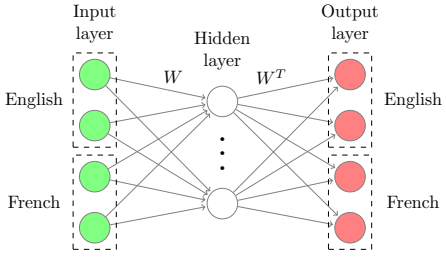

Fig. 1: An AE that generates the PE in its hidden layer. The dashed boxes denote the document DRs in the corresponding language.

	Distributed Representations			Classification			
	Documents	Vocabulary	# Words	Documents	Vocabulary	Avg. Doc. Len	# Labels
English	6,358,467	490,122	198,213,780	12,670	56,886	115.32	1,17
French	6,358,467	713,171	177,766,544	12,670	58,678	132.29	1,17

Table 1: Statistics after pre-processing the datasets. The distributed representations dataset refers to the data used to train $\mathcal{G}$. The classification data refer to the supervised dataset used for classification purposes.

3 The Experimental Framework

The data. Training neural network models to generate distributed representations benefits by large amounts of free text. To train the models that generate DRs we used such free texts in English and French:[3] the left part of Table 1 (under "Distributed Representations") presents some basic statistics for those data. We used the same number of documents for the two languages to avoid any training bias. The raw text was pre-processed by applying lower-casing and space-padding punctuation. Similarly to previous studies [8,7], we kept the punctuation. Publicly available implementations of the models were used with their default parameters: the word2vec tool[4] for the cbow and skip-gram and the doc2vec for the DBOWpv and DMMpv from Gensim [11].

For the classification task we used the raw version of the Wikipedia dataset of the Large Scale Hierarchical Text Classification challenge [10]. The original dataset contains 60,252 categories; we restrict our study here in a fraction of the dataset with 12,670 documents belonging to the 100 most common categories. The right part of Table 1 presents basic statistics for this subset.

Baselines. We used as a first baseline Support Vectors Machines (SVM) fed with the tf-idf representation of the documents, which is commonly used in text classification problems (denoted by SVM_{BoW}). As a second baseline, we used k-Nearest Neighbours (k-NN) and SVMs learned on the monolingual space of the DRs of English documents (denoted respectively by SVM_{DR} and $k\text{-NN}_{\text{DR}}$). These baselines aim at evaluating the value of the fusion mechanism (PE) that we propose. k-NN and SVMs were adapted to the multi-label setting (denoted respectively by SVM_{PE} and $k\text{-NN}_{\text{PE}}$). For the former, given the labels of the k nearest training instances of a test document, the algorithm returns the labels that belong to at least p% of its nearest neighbours. For each run $k \in \{13, 14, 15\}$ and $p \in \{0.1, 0.2, 0.3\}$ are decided using 5-fold cross-validation on the training data. The SVMs were used in an one-vs-rest fashion; they return every label that has a positive distance from the separating hyperplane. The value of the hyper-parameter $C \in \{10^{-1}, \ldots, 10^4\}$ that controls the importance of the regularization term in the optimization problem, is selected using 5-fold cross-validation over the training data.

[3] `http://statmt.org/`

[4] `https://code.google.com/p/word2vec/`

	cbow				skip-gram			
dim.	k-NN$_{\text{DR}}$	SVM$_{\text{DR}}$	k-NN$_{\text{PE}}$	SVM$_{\text{PE}}$	k-NN$_{\text{DR}}$	SVM$_{\text{DR}}$	k-NN$_{\text{PE}}$	SVM$_{\text{PE}}$
50	39.19	37.20	39.58	32.84	38.25	34.74	37.51	32.09
100	40.20	40.01	43.53	37.54	39.34	38.61	41.15	34.54
200	40.48	43.41	45.86	42.50	39.73	40.96	42.79	41.08
300	40.42	44.25	**46.33**	43.38	39.62	42.67	42.62	42.74
	DBOWpv				DMMpv			
50	24.45	25.06	30.26	24.08	24.47	25.56	29.55	24.94
100	31.20	28.53	34.63	26.88	24.74	29.31	31.21	27.22
200	27.73	29.80	36.02	30.80	18.22	30.04	29.01	32.10
300	27.79	29.92	38.71	30.82	15.98	30.49	25.20	32.01
SVM$_{\text{BoW}}$	36.03							

Table 2: F_1 measures of difference algorithms. The performance of 5-fold cross-validated SVM using the bag-of-words representation is 36.03

Our approach. Using the above presented DR model, we first generated the document embeddings in English and French in a d-dimensional space with $d \in \{50, 100, 200, 300\}$. Then, for the AE we considered as activation functions the hyperbolic tangent and the sigmoid function. The sigmoid performed consistently better and thus we use it in the reported results. The AE was trained with tied weights using a stochastic back-propagation algorithm with mini-batchs of size 10 and the euclidean distance of the input/output as loss function. The number of neurons in the hidden layer was set to be 70% of the size of the input.[5]

4 Experimental Results

Table 2 presents the scores of the F_1 measure when 10% of the 12.670 documents were used for training purposes and the rest 90% for testing. We report the classification performance with the four different DR models (cbow, skip-gram, DBOWpv and DMMpv) and 2 learning algorithms (k-NN and SVMs) for different input sizes. The columns labeled k-NN$_{\text{DR}}$ and SVM$_{\text{DR}}$ present the (baseline) performance of SVM and k-NN trained on the monolingual DRs. Also the last line of the table indicates the F_1 score of SVM with tf-idf representation (SVM$_{\text{BoW}}$). The best obtained result is shown in bold.

We first notice that the average pooling strategy (cbow and skip-gram) performs better compared to when the document vectors are directly learned (DBOWpv and DMMpv). In particular, cbow seems to be the best performing representation, both as a baseline model and when used as base model to generate the PE representations. On the other hand, DBOWpv and DMMpv perform significantly worse: in the baseline setting the best cbow performance achieved is 44.25 whereas the best DMMpv configuration achieves 30.49, 14 F_1 points less.

The PE representations learned on top of the four base models improve significantly over the performance of the monolingual DRs, especially for k-NN.

[5] The code is available at `http://ama.liglab.fr/~balikas/ecir2015.zip`.

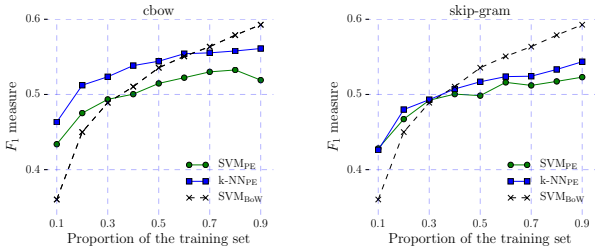

Fig. 2: Comparison of the performance of the learning algorithms learned on different representations with respect to the available labelled data. The dimension of the PE representations is 300.

For instance, for cbow with base-model vector dimension 200, the baseline representation achieves 40.42 F_1 and its corresponding PE representation obtains 46.33, improving almost 6 points. In general, we notice such improvements between the base DR and its respective PE, especially when the dimension of the DR representation increases. Note that the PE improvements are independent of the methods used to generate the DRs: for instance k-NN$_{\text{PE}}$ over the 200-dimensional PE DMMpv representations gains more than 11 F_1 points compared to k-NN$_{\text{DR}}$. It is also to be noted that the baseline SVM$_{\text{BoW}}$ is outperformed by SVM$_{\text{PE}}$ especially when cbow and skip-gram DRs are used.

Comparing the two learning methods (k-NN$_{\text{PE}}$ and SVM$_{\text{PE}}$), we notice that k-NN$_{\text{PE}}$ performs best. This is motivated by the fact that distributed representations are supposed to capture the semantics in the low dimensional space. At the same time, the neighbours algorithm compares exactly this semantic distance between data instances, whereas SVMs tries to draw separating hyperplanes among them. Finally, it is known that SVMs benefit from high-dimensional vectors such as bag-of-words representations. Notably, in our experiments increasing the dimension of the representations consistently benefits SVMs.

We now examine the performance of the PE representations taking into account the amount of labeled training data. Figure 2 illustrates the performance of the SVM$_{\text{BoW}}$ and SVM$_{\text{PE}}$ and k-NN$_{\text{PE}}$ with PE representations when the fraction of the available training data varies from 10% of the intial training set to 90% and in the case where, cbow and skip-gram are used as DR representations with an input size of 300. Note that if only a few training documents are available, the learning approach is strongly benefited by the rich PE representations, that outperforms the traditional SVM$_{\text{BoW}}$ setting consistently. For instance, in the experiments with 300 dimensional PE representations with cbow DRs, when only 20% of the data are labeled, the SVM$_{\text{BoW}}$ needs 20% more data to achieve similar performance, a pattern that is observed in most of the runs in the figure. When, however, more training data are available the tf-idf copes with the complexity of the problem and levarages this wealth of information more efficiently than PE does.

5 Conclusion

We proposed the PE, which is a text embedding learned using neural networks by leveraging translations of the input text. We empirically showed the effectiveness of the bilingual embedding for classification especially in the interesting case where few labeled training data are available for learning.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This work is partially supported by the CIFRE N 28/2015 and by the LabEx PERSYVAL Lab ANR-11-LABX-0025.

References

1. Faruqui, M., Dyer, C.: Improving vector space word representations using multilingual correlation. Association for Computational Linguistics (2014)
2. Gao, J., He, X., Yih, W.t., Deng, L.: Learning continuous phrase representations for translation modeling. Proc. of ACL. Association for Computational Linguistics, June (2014)
3. Gouws, S., Bengio, Y., Corrado, G.: Bilbowa: Fast bilingual distributed representations without word alignments. arXiv preprint arXiv:1410.2455 (2014)
4. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
5. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
6. Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V.C., Saha, A.: An autoencoder approach to learning bilingual word representations. In: Advances in Neural Information Processing Systems. pp. 1853–1861 (2014)
7. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. arXiv preprint arXiv:1405.4053 (2014)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119 (2013)
10. Partalas, I., Kosmopoulos, A., Baskiotis, N., Artieres, T., Paliouras, G., Gaussier, E., Androutsopoulos, I., Amini, M.R., Galinari, P.: Lshtc: A benchmark for large-scale text classification. arXiv preprint arXiv:1503.08581 (2015)
11. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), `http://is.muni.cz/publication/884893/en`
12. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 1555–1565 (2014)
13. Zhang, X., LeCun, Y.: Text understanding from scratch. arXiv preprint arXiv:1502.01710 (2015)