Generalized Funnelling: Ensemble Learning and Heterogeneous Document Embeddings for Cross-Lingual Text Classification

Discussion Paper

Alejandro Moreo[1], Andrea Pedrotti[1,2] and Fabrizio Sebastiani[1]

[1]*Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy*
[2]*Dipartimento di Informatica, Università di Pisa, 56127 Pisa, Italy*

Abstract

Funnelling (Fun) is a method for cross-lingual text classification (CLTC) based on a two-tier learning ensemble for heterogeneous transfer learning (HTL). In this ensemble method, 1st-tier classifiers, each working on a different and language-dependent feature space, return a vector of calibrated posterior probabilities (with one dimension for each class) for each document, and the final classification decision is taken by a metaclassifier that uses this vector as its input. In this paper we describe *Generalized Funnelling* (gFun), a generalization of Fun consisting of a HTL architecture in which 1st-tier components can be arbitrary *view-generating functions*, i.e., language-dependent functions that each produce a language-independent representation ("view") of the document. We describe an instance of gFun in which the metaclassifier receives as input a vector of calibrated posterior probabilities (as in Fun) aggregated to other embedded representations that embody other types of correlations. We describe preliminary results that we have obtained on a large standard dataset for multilingual multilabel text classification.

Keywords

Transfer Learning, Cross-Lingual Text Classification, Ensemble Learning, Word Embeddings

1. Introduction

According to [1], the amount of (labelled and unlabelled) resources for the more than 7,000 languages spoken around the world follows (somehow unsurprisingly) a power-law distribution. That is, while a small set of languages account for most of the available data, a very long tail of other languages suffer from data scarcity, despite the fact that many languages belonging to this long tail have large speaker bases.

Bearing in mind that most of the languages in the world are low-resource, it is appealing to develop methods and techniques capable of exploiting the high-quality resources available for the few resource-rich languages, in order to improve the performance on tasks carried out on the resource-poor languages. *Cross-Lingual Transfer Learning* (CLTL) is a class of machine learning tasks in which, given a training set of textual labelled data sampled from one or more

IIR 2021 – 11th Italian Information Retrieval Workshop, September 13–15, 2021, Bari, Italy
alejandro.moreo@isti.cnr.it (A. Moreo); andrea.pedrotti@phd.unipi.it (A. Pedrotti); fabrizio.sebastiani@isti.cnr.it (F. Sebastiani)
0000-0002-0377-1025 (A. Moreo); 0000-0002-2322-7043 (A. Pedrotti); 0000-0003-4221-6427 (F. Sebastiani)

source languages, we must issue predictions for unlabelled documents written in one or more *target* languages. In other words, the goal of CLTL is to transfer (i.e., reuse) the knowledge that has been obtained from the training data in the source languages, to the target languages of interest, for which few labelled data (or no labelled data at all) exist.

Cross-Lingual Text Classification (CLTC) is a specific instance of CLTL, in which classification is the task to be carried out. In CLTC, documents are written in one of a finite set $\mathscr{L} = \{\lambda_1, ..., \lambda_{|\mathscr{L}|}\}$ of languages, and labelled according to a shared *codeframe* (a.k.a. *classification scheme*) $\mathscr{Y} = \{y_1, ..., y_{|\mathscr{Y}|}\}$. In such a scenario, it is common to have different numbers of training documents for the different languages, with the languages with fewer training documents usually being also the ones with fewer (if at all) available external resources (such as bilingual dictionaries, thesauri, pre-trained sets of word-embeddings, language models) that could otherwise be leveraged for this task.

Funnelling (Fun – [2]) is an ensemble learning architecture for CLTC especially designed to learn from heterogeneous sources of data and effectively transfer information from one language to another. In other words, Fun operates in an *all-to-all* fashion since all training languages contribute to the classification of the other languages while, at the same time, all languages benefit from the training data which is available for other languages. In this work we expand over this architecture by injecting into the algorithm *new* heterogeneous sources of information.

2. Funnelling and Generalized Funnelling

Fun is a two-level architecture [2], where the first tier takes care of translating documents from their original language-dependent domain to a language-independent one. Subsequently, the second tier operates on the newly encoded documents and outputs the final prediction scores.

The main intuition behind Fun is to leverage the fact that all the documents are classified according to the same set of labels. Documents, regardless of the language they are written in, can be represented as vectors of posterior probabilities, i.e., vectors encoding, at each dimension i, the probability for a given document to be labeled as belonging to the respective class y_i. Once all the documents are homogenized (i.e., they are all represented as vectors of posterior probabilities), they can be stacked vertically and fed to the second-tier (the metaclassifier) regardless of the language they were originaly written in.

We generalize this architecture, and call it Generalized Funnelling (gFun). The first tier of Fun is redesigned in order to accommodate for a a set Ψ of *view-generating functions* (VGFs) that can expand the shared vector space on which the meta-classifier operates. VGFs are language-dependent functions that map documents into language-independent vectorial representations (*views*) aligned across languages. Since each view is aligned across languages, it is easy to aggregate (e.g., by concatenation) the different views into a single representation aligned across languages, that is then given as input to the metaclassifier. Notice that, according to this definition, the original implementation of Fun can be seen as a specific setting of gFun equipped with one single VGF.

The key idea is to leverage the VGFs in order to inject into the model information about different correlations between the main elements of a Text Classification task.In this research,

we consider four kinds of correlations: *Class-Class correlation, Document-Class correlation, Word-Class correlation, Word-Word correlation, Document-Word correlation.* We bring to bear these stochastic correlations by means of the following VGFs:

- the *Posteriors VGF* (encoding document-class correlations): it maps documents into the space defined by calibrated posterior probabilities (as in the original Fun).
- the *MUSEs VGF* (encoding word-word correlations): it uses the Multilingual Unsupervised / Supervised Embeddings (MUSEs) made available by the authors of [3], a set of word embeddings aligned for 30 languages.
- the *WCEs VGF* (encoding word-class correlations): it uses Word-Class Embeddings (WCE) [4], a form of supervised word embeddings based on the class-conditional distributions observed in the training set which is natively aligned across languages.
- the *BERT VGF* (encoding document-word correlations): it uses the contextualized word-embeddings generated by multilingual BERT [5], a deep pretrained language model based on the transformer architecture.

The different views produced by the VGFs need to be aggregated before being issued to the metaclassifier. In this work, we propose to *average* the different views.[1] Before averaging the representations, we must ensure all views to have same dimensionality, and to be aligned.[2] In order to do so, we learn additional mappings of the views to the space of class-conditional posterior probabilities, i.e., for each VGF (other than the Posteriors VGF, which already returns vectors of $|\mathcal{Y}|$ calibrated posterior probabilities) we train a classifier that maps the view of a document into a vector of $|\mathcal{Y}|$ calibrated posterior probabilities.

Finally, we have found that applying some routine normalization techniques consistently increases the performance of gFun. This normalization consists of imposing unit L2-norm to the vectors computed by the view generators, removing the first principal component of the document embeddings obtained via WCEs or MUSEs, and standardizing the columns of the shared space before passing the vectors [6] to the metaclassifier.[3]

3. Experiments

In order to maximize comparability with the previous results, we adopt an experimental setup identical to the one used in [2] including the evaluation metrics, i.e., F_1 score and K, in both their micro (μ) and macro-averaged (M) versions.

We carry out experiments on JRC-Acquis, a parallel corpus of legislative texts published by the European Union, consisting of 11 different languages. We retain the 300 most frequent

[1] In preliminary work, we have observed experimentally that avaraging tends to produce better results than simply concatenating the different views.

[2] Two views are said to be aligned when the semantics of the dimensions (whatever it may be) is common to both views.

[3] Standardizing (a.k.a. "z-scoring", or "z-transforming") consists of having a random variable x, with mean μ and standard deviation σ, translated and scaled as $z = \frac{x-\mu}{\sigma}$, so that the new random variable z has zero mean and unit variance. The statistics μ and σ are unknown, and are thus estimated on the training set.

target classes and use the same splits as in [2].[4]

In Table 1, we directly compare our results with the naïve solution (i.e., one monolingual classifier for each language), Fun and multilingual BERT (mBERT). We group gFun results in three different batches: the first one groups the results obtained by deploying one single VGF at the time; in the second one we report the results combining multiple generators; in the latter we deployed all the proposed VGFs jointly. We use the notation -X to refer to the Posteriors VGF, -M denotes the MUSEs VGF, -W the WCEs VGF, and -B the BERT VGF.

The superior results of gFun-Xwith respect to Fun indicate that the normalization steps are beneficial. It is noteworthy how by simply leveraging the class-class correlations (brought to bear by the metaclassifier) gFun-B outperforms its counterpart mBERT. The best results are obtained by the combination of Posterior, MUSEs, and BERT VGFs.

Table 1
CLTC results on JRC-Acquis dataset. Each cell indicates the mean value and the standard deviation across the 10 runs. **Boldface** indicates the best method. Superscripts † and †† denote the method (if any) whose score is not statistically significantly different from the best one;

Method	F_1^M	F_1^μ	K^M	K^μ
Naïve	.340 ± .017	.559 ± .012	.288 ± .016	.429 ± .015
Fun [2]	.399 ± .013	.587 ± .009	.365 ± .014	.490 ± .013
mBERT [5]	.420 ± .023	.608 ± .016	.379 ± .006	.507 ± .009
gFun–X	.432 ± .015	.587 ± .010	.441 ± .016	.553 ± .013
gFun–M	.440 ± .039	.586 ± .032	.442 ± .045	.549 ± .034
gFun–W	.410 ± .016	.553 ± .014	.410 ± .021	.525 ± .022
gFun–B	.501 ± .023	.627 ± .016	.485 ± .023	.574 ± .019
gFun–XB	.510 ± .017	.637 ± .012	.512 ± .020†	.603 ± .016†
gFun–XMB	**.525** ± .020	**.649** ± .014	**.528** ± .023	**.620** ± .017
gFun–XWB	.497 ± .011	.621 ± .008	.508 ± .011	.606 ± .010
gFun–XMW	.475 ± .012	.604 ± .010	.489 ± .014	.593 ± .011
gFun–WMB	.513 ± .016	.632 ± .011	.522 ± .017††	.619 ± .013††
gFun–XWMB	.514 ± .014	.635 ± .010	.521 ± .015†	.618 ± .011††
UPPERBOUND	.599	.707	.547	.632

4. Conclusions

In this paper we propose Generalized Funnelling (gFun), a revised variant of Fun [2] that allows a set of *view-generating functions* (VGFs) to provide the metaclassifier with different views of the same document, each embodying a different type of correlation in the data. We explore views leveraging the *multilingual unsupervised-supervised embeddings* (MUSE) [3], *word-class embeddings* (WCE) [4], and the contextualized embeddings of multilingual BERT [5]. The results confirm that injecting in the process heterogeneous information in the form of different types of embeddings aligned across languages improves performance in CLTL.

[4]We have validated our method also using RCV1/2, but we leave the discussion of this dataset out of this short paper for the sake of brevity.

References

[1] P. Joshi, S. Santy, A. Budhiraja, K. Bali, M. Choudhury, The state and fate of linguistic diversity and inclusion in the NLP world, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), 2020, pp. 6282–6293. doi:10.18653/v1/2020.acl-main.560.

[2] A. Esuli, A. Moreo, F. Sebastiani, Funnelling: A new ensemble method for heterogeneous transfer learning and its application to cross-lingual text classification, ACM Transactions on Information Systems 37 (2019) Article 37. doi:https://doi.org/10.1145/3326065.

[3] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, H. Jégou, Word translation without parallel data, in: Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, CA, 2018.

[4] A. Moreo, A. Esuli, F. Sebastiani, Word-class embeddings for multiclass text classification, Data Mining and Knowledge Discovery 353 (2021) 911–963. doi:10.1007/s10618-020-00735-3.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2019), Minneapolis, US, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[6] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, FR, 2017.