On the evaluation of retrofitting for supervised short-text classification[1]

Kaoutar GHAZI [a], Andon TCHECHMEDJIEV [a], Sébastien HARISPE [a], Nicolas SUTTON-CHARANI [a] and Gildas TAGNY NGOMPÉ [b]

[a] *EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Alès, Alès, France*
[b] *ESII, Laverune - France*

Abstract. Current NLP systems heavily rely on embedding techniques that are used to automatically encode relevant information about linguistic entities of interest (e.g., words, sentences) into latent spaces. These embeddings are currently the cornerstone of the best machine learning systems used in a large variety of problems such as text classification. Interestingly, state-of-the-art embeddings are commonly only computed using large corpora, and generally do not use additional knowledge expressed into established knowledge resources (e.g. WordNet). In this paper, we empirically study if retrofitting, a class of techniques used to update word vectors in a way that takes into account knowledge expressed in knowledge resources, is beneficial for short text classification. To this aim, we compared the performances of several state-of-the-art classification techniques with or without retrofitting on a selection of benchmarks. Our results show that the retrofitting approach is beneficial for some classifiers settings and only for datasets that share a similar domain to the semantic lexicon used for the retrofitting.

Keywords. text classification, word embeddings, retrofitting

Introduction

Embedding techniques are the cornerstone of numerous state-of-the-art NLP systems; they enable to automatically encode relevant information about linguistic entities of interest (e.g., words, sentences, documents) into latent spaces in order to obtain high quality representations that will further be used to solve complex tasks. Such techniques have proven to be critical for designing efficient systems in text classification [1], question answering [2] or information extraction [3] to mention a few.

Neural network architectures, particularly recurrent neural networks (RNN) or Transformers are now *de facto* approaches to computing embeddings, as illustrated by the broad variety of language models of increasing complexity and efficiency that have been published in recent years (e.g. RoBERTa [4], GPT-3 [5]). These approaches rely on the surface analysis of large corpora composed of billions of words, and do not use additional knowledge expressed into established knowledge resources (e.g. WordNet). Despite recent successes, there is only so much that can be learned from a surface analy-

sis of text and embedding models capture very superficial knowledge about meaning [6]. One way of integrating structured *a priori* knowledge, is to apply *retrofitting*, a class of techniques used to update word vectors in a way that takes into account knowledge expressed in knowledge resources. Despite the promising results obtained by retrofitting techniques, the study of hybrid embedding approaches mixing corpora and knowledge representations is still relatively marginal, especially in context of specific tasks. This paper aims at investigating the relevance of retrofitted word embeddings in the context of supervised short-text classification, especially when compared to state of the art contextualised language models. We compare the performance of several pre-trained word embedding models with and without retrofitting for short-text classification. We explored several retrofitting approaches and used word vectors as features with both a classical machine-learning pipeline and a more state of the art bi-LSTM encoder. We further compared to two transformer baselines, where the transformers are directly used for classification.

The paper is organized as follows: Section 1 briefly present the tow most common retrofitting models; Section 2 presents the protocol used in our experimental setting as well as the obtained results. Section 3 discusses those results and offers additional observations that question the benefit of current retrofitting approaches for the studied task.

1. Retroffiting embeddings in NLP

State-of-the-art word embedding techniques solely based on corpora analysis are performed under the assumption of the distributional hypothesis stating that words occurring in similar contexts tend to be semantically close [7]. This hypothesis, made popular through Firth's idea (1957) [8]: "*You shall know a word by the company it keeps*", is one of the main tenets of statistical semantics. By definition such approaches cannot capture lexical or conceptual relationships that could be important to accurately characterizing the semantics of words, e.g. some approaches will similarly represent synonyms and antonyms [9]. To address this limitation, a class of approaches denoted *Retrofitting* aim at incorporating *a priori* knowledge from external resources in order to refine word embeddings, e.g. lexicons, ontologies, domain-specific datasets expressing semantic knowledge.

The use of external data or knowledge generally requires retraining the model used to compute the embeddings (considered as a subset of the model's parameters). In this case retrofitting can be seen as a post-processing step that aims at updating pre-trained word embeddings in order to induce a refined vector space with desired properties encoded in the external resource. Indeed, in addition to observed words contexts (i.e. surroundings), resources such as semantic lexicons (such as FrameNet, PPDB and WordNet) that label lexical entries with semantic relations (e.g. hyperonymy, hyponymy) can be used. In the literature, the prevailing approach is to define a specific objective function that learns the distribution of words and their lexical (resp. conceptual) relationships either jointly [10,11,12,13], or separately by updating pre-trained embeddings [14,15]. When embeddings are updated using lexical ontologies, the objective function depends on which semantic relation we seek to highlight: synonymy, hyperonymy, hypernymy (the "retrofitting" technique) [14] or synonymy and antonymy (the "counterfitting" technique) [15]. These approaches are particularly interesting as they can be applied to any

word embeddings independently from the embeddings technique initially used to generate them.

Another strategy learns independent representations from corpora and knowledge representations to later combine them. For instance, Goikoetxea et al. [16] learned word representations from WordNet, and combined them with embeddings computed from text. Several contributions have been proposed to refine these general strategies, e.g. Vulić et al. [17] have proposed an approach based on context analysis enabling to retrofit words that do not occur in the lexicon by exploiting words co-occurring in similar dependency-based contexts; Yih et al. [9] proposed to use a thesaurus to distinguish synonyms from antonyms in word embeddings.

These approaches have traditionally been proposed for static word representations, i.e. a single representation is associated to a word (token). Recently, contextualized text-embedding models have been proposed to deal with issues induced by polysemy [18,19]. In this case, a context-specific representation of a word is obtained depending on its meaning in each sentence. Recent retrofitting techniques are designed for these contextualized embeddings, e.g. Shi et al. [20] proposed to consider prior knowledge about paraphrases to improve context-specific representations.

Several studies have stressed the benefits of retrofitting on several NLP tasks such as sentiment analysis, relationship and text classification [14,15,21,9,17,10,11,12]. These studies only contain limited comparisons with state-of-art language models, in particular for short-text classification.

2. Evaluation protocol

This section presents the datasets and protocol used to evaluate the benefit of retrofitting approaches for short-text classification. We focus our study on the following well-known and representative retrofitting techniques: "retrofitting" of Faruqui et al. [14] and "counterfitting" of Mrkšić et al. [15].

2.1. Word Embeddings

we considered the 300-dimensional word vectors: (i) Paragram [16], learned from the text content in the paraphrase database PPDB, (ii) Glove [22] learned from Wikipedia and Common Crawl data, (iii) MUSE, a fastText embedding learned from Wikipedia[2], as well as (iv) two contextualized word embeddings models: Flair embeddings [23] trained on JW300 corpus, and RoBERTa [4] embeddings trained on five English corpora: BookCorpus [24]; Wikipedia; CC-NEWS [25]; Open Web Text [26] and Stories [27].

For each word embedding model, except for the contextualized embedding baselines, we consider three settings: original embeddings (baseline), retrofitted and counterfitted.

2.2. Evaluation benchmarks

The evaluations were performed on two benchmarks:

[2] `https://github.com/facebookresearch/MUSE`

- *HuffPost headlines*[3] [28]: 200849 headlines published in HuffPost from 2012 to 2018. Each news headline belongs to one of 41 possible classes.
- *Product Listing On Amazon India*[4]: 27375 product titles from Amazon India for 2019. We keep the products belonging one of the 9 classes and redundant records have been dropped.

2.3. Evaluation Process

We consider two different evaluation settings: (i) *shallow machine learning* where we compute a single document vector by pooling individual word embeddings, which we use as a bag of features for several classifiers; (ii) *deep machine learning*, where we use a bi-LSTM encoder [23] to learn document representations from word embeddings during the training of a final feed forward layer. Pre-transformer literature suggests that the ability of LSTM to capture dependencies between words, makes it a robust choice for text classification applications.

In the first context, three models are compared: the ridge classifier, random forest and XGBoost from scikit-learn [29]. In the second context we use Flair embeddings [23] with its RNN Document Embedding implementation initialised with bi-LSTM cells for each model. We apply a grid search on held-out training data to find the best hyper-parameter values and then we run a 10-folds cross validation considering the optimally hyper-paremeters for each model. Words embeddings (baseline, retrofitted or counterfitted) are given as input for each model. In the shallow setting we compute pooled document vectors with flair's document pool embedding implementation (mean pooling with a linear smoothing); in the deep setting unpooled word embeddings are given as input to the LSTM encoder. In addition, we also present baselines using embeddings from Flair RoBerta in both settings, as well as a direct classification with the Transformer model with a classification head (using RoBerta).

3. Results and discussion

Table 1 reports the average accuracies over the 10 cross-validation folds for all models on the two benchmarks. Results are grouped depending on the embedding used in the tested approach. It is important to highlight the impact of retrofitting on the performance with relation to the corresponding baseline approach, i.e. considering the use of the original embedding without retrofitting [5]. We also report the accuracy delta compared to the corresponding baseline accuracy.[6] In the shallow setting, we only reported on the best performing classifier (always the ridge classifier). For the LSTM-RNN approach, the standard deviations of the averaged accuracies obtained during cross-validation are generally around $1-5\%$ for the Huffington post dataset and $5-12\%$ for the Amazon India dataset. For the Ridge approach, the standard variations are always under 1%.

[3] `https://www.kaggle.com`

[4] `https://data.world/promptcloud/product-listing-on-amazon-india`

[5] We draw the reader's attention to the fact that the embedding models considered (Paragram, Muse, Glove...) have not all been trained on the same corpora.

[6] e.g.For the HuffPost dataset, Paragram embbedings retrofitted with PPDB leads to an accuracy of 42.80% with the ridge classifier, which corresponds to a 0.03% accuracy improvement compared to the Paragram baseline (42.77%).

Embeddings	Semantic Lexicon	HuffPost Headlines		Product Amazon India	
		RidgeC	LSTM-RNN	RidgeC	LSTM-RNN
Paragram	∅	42.77	63.87	36.67	**62.49**
Paragram Retrofitted	PPDB	42.80 (+0.03)	**66.78** (+2.91)	36.63 (-0.04)	53.24 (-9.25)
	FrameNet	42.57 (-0.20)	57.82 (-6.05)	36.39 (-0.28)	44.68 (-17.81)
	WordNet$_{syn}$	42.63 (-0.14)	59.83 (-4.04)	36.56 (-0.11)	53.37 (-9.12)
	WordNet+	42.38 (-0.39)	60.44 (-3.43)	36.64 (-0.03)	46.38 (-16.11)
Paragram Counterfitted	PPDB&WordNet	42.57 (-0.20)	59.33 (-4.54)	35.97 (-0.70)	48.12 (-14.37)
Glove	∅	45.28	65.06	36.94	59.51
Glove Retrofitted	PPDB	45.40 (+0.12)	59.21 (-5.85)	36.84 (-0.10)	59.25 (-0.26)
	FrameNet	44.89 (-0.39)	63.08 (-1.98)	36.87 (-0.07)	48.56 (-10.95)
	WordNet$_{syn}$	44.69 (-0.59)	63.80 (-1.26)	36.54 (-0.40)	56.24 (-3.27)
	WordNet+	44.52 (-0.76)	66.65 (+1.59)	36.56 (-0.38)	42.25 (-17.26)
Glove Counterfitted	PPDB&WordNet	44.32 (-0.96)	55.83 (-9.23)	36.63 (-0.31)	48.67 (-10.84)
MUSE	∅	44.80	59.94	36.26	48.87
MUSE Retrofitted	PPDB	45.20 (+0.40)	64.92 (+4.98)	36.25 (-0.01)	35.42 (-13.4)
	FrameNet	44.43 (-0.37)	64.64 (+4.70)	35.87 (-0.39)	58.71 (+9.84)
	WordNet$_{syn}$	44.27 (-0.53)	58.44 (-1.50)	36.40 (-0.14)	38.91 (-9.96)
	WordNet+	44.20 (-0.60)	64.85 (+4.91)	36.10 (-0.16)	56.81 (+7.94)
MUSE Counterfitted	PPDB&WordNet	44.07 (-0.73)	66.41 (+6.47)	36.01 (-0.25)	51.29 (+2.42)

Table 1. Accuracy (%) of the classification for the ridge and LSTM-RNN classifiers with baseline word embeddings (no retrofitting), retrofitted and counterfitted word embeddings.

The best performances for HuffPost Headlines are achieved with the LSTM-RNN approach using embeddings refined by retrofitting (maximum average accuracy of 66.78%). For the Amazon India dataset the best average accuracy is 62.49% in the baseline Paragram setting. In the shallow machine learning setting, we can hardly observe any improvement with retrofitting (variations too small to be significant), which can be attributed to the hypothesis that a linear classifier cannot meaningfully capture the additional information. The impact of retrofitting is clearer on LSTM-RNN, although we observe large variations of the average accuracy and a higher overall variability across folds. Since the LSTM-RNN encoder is trained alongside classification layer, we effectively learn a non-linear supervised document representations that can both capture some dependencies and map the original feature space in a meaningful way. The improvements mainly concern the HuffPost Headlines dataset (news domain). Given that some of the embeddings are trained on news corpora and that the lexicons used for retrofitting mostly (except PPDB) cover the general domain, it is reasonable to assume that the retrofitting mostly benefits data in the same domain. For example, retrofitting Paragram embeddings with PPDB leads to a +2.91% average accuracy improvement using an LSTM-RNN classifier on HuffPost headlines; the same approach applied to Product Amazon India leads to a 9.25% decrease of the average accuracy. Generally, the results underline the difficulty of formulating recommendations for one particular approach. However, we can identify that MUSE most often benefits from retrofitting than not. Compared to other

words embeddings concidered, MUSE embeddings are learned from the smallest and the most general corpus (Wikipedia).

The impact of the corpus used for computing words vectors is also emphasized by the evaluation on contextualized embeddings. In fact, we also evaluated Flair and RoBERTa embeddings as features with the ridge classifier for both data-sets [7], obtaining an accuracy of 53.48% (resp. 39.24%) with RoBERTa, and only 41.81% with Flair embeddings (resp. 34.69%) for HuffPost Headlines (resp. Product Amazon India): better initial representations lead to a better classification result even with an unsophisticated classifier. With less meaningful input representation it 's beneficial to have some form of task-specific representation learning to help the classifier exploit all meaningful information in the features. We also tested Flair and RoBerta with a LSTM-RNN head, however the significantly larger number of parameters did not allow the models to converge with similar computational constraints to the other models [8].

4. Conclusion

By definition, embedding techniques only based on corpora analysis are not designed to capture lexical or conceptual relationships that could be important to accurately characterizing the semantics of words. To address this limitation, a class of approaches denoted *Retrofitting* has been proposed in the litterature to incorporate *a priori* knowledge expressed into knowledge resources, e.g. lexical ontologies. Questioning the benefit of such approaches requires extensive task-specific empirical evaluations.

In this context, this paper presents an evaluation of the impact of state-of-the-art retrofitting approaches for short-text classification using shallow and deep learning models on two datasets: HuffPost Headlines and Product Amazon India. Two retrofitting techniques of interest, as they enable refinement of existing embeddings, have been tested using several external resources. The baseline retrofitting used a single ontology (i.e. PPDB) that captures similar words while the counterfitting technique used two external resources that captures similar and dissimilar words respectively. We applied these techniques on several pre-trained words embeddings. We compared retrofitted and counterfitted embeddings with contextualized ones. Based on the results obtained in our evaluation, we conclude that current retrofitting techniques generally fail to systematically and significantly improve classification performance. Indeed, despite interesting gains using some configurations (retrofitting technique, resource and classification method), no general tendency and recommendations can be expressed. Tested shallow machine learning models seem not to benefit from retrofitting; Deep Learning approaches such as LSTM-RNN do in some settings: interesting gains have been observed using Paragram embeddings with PPDB, or MUSE embeddings with PPDB, FrameNet or WordNet+ for HuffPost Headlines dataset (same domain), for the Amazon India dataset we saw little benefit to using retrofitting (different domain).

In future work, we can explore the retrofitting approach for contextualized word embeddings as proposed by Shi et al. in [20]. We can also use all semantic lexicons together

[7] Equivalent to the transformer with a classification head and frozen weights

[8] For retrofitted embeddings + Ridge training and evaluation were almost instantaneous. For retrofitted embedding + LSTM-RNN (20 epochs) we had approx. 1000 samples/s, for RoBERTa, 29 samples/s.

to retrofit each embeddings or use domain-specific lexical ontologies or terminologies for the retrofitting.

References

[1] Peng Jin, Yue Zhang, Xingyuan Chen, and Yunqing Xia. Bag-of-embeddings for text classification. In IJCAI, volume 16, pages 2824–2830, 2016.

[2] Guangyou Zhou, Tingting He, Jun Zhao, and Po Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 250–259, 2015.

[3] Xin Ye, Hui Shen, Xiao Ma, Razvan Bunescu, and Chang Liu. From word embeddings to document similarities for improved information retrieval in software engineering. In Proceedings of the 38th international conference on software engineering, pages 404–415, 2016.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.

[6] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In Proc. of ACL, 2020.

[7] Zellig S Harris. Distributional structure. Word, 10(2-3):146–162, 1954.

[8] J.R. Firth. Studies in Linguistic Analysis: Special Volume of the Philosogical Society. Special Volume of the Philological Society. Blackwell, 1957.

[9] Wen-tau Yih, Geoffrey Zweig, and John C Platt. Polarity inducing latent semantic analysis. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1212–1222, 2012.

[10] Mo Yu and Mark Dredze. Improving lexical embeddings with semantic knowledge. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 545–550, 2014.

[11] Jiang Bian, Bin Gao, and Tie-Yan Liu. Knowledge-powered deep learning for word embedding. In Joint European conference on machine learning and knowledge discovery in databases, pages 132–148. Springer, 2014.

[12] Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. Rc-net: A general framework for incorporating knowledge into word representations. In Proceedings of the 23rd ACM international conference on conference on information and knowledge management, pages 1219–1228, 2014.

[13] Daniel Fried and Kevin Duh. Incorporating both distributional and relational semantics in word representations. arXiv preprint arXiv:1412.4369, 2014.

[14] Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. arXiv preprint arXiv:1411.4166, 2014.

[15] Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. arXiv preprint arXiv:1603.00892, 2016.

[16] Josu Goikoetxea, Eneko Agirre, and Aitor Soroa. Single or multiple? combining word representations independently learned from text and wordnet. In AAAI, pages 2608–2614, 2016.

[17] Ivan Vulić, Roy Schwartz, Ari Rappoport, Roi Reichart, and Anna Korhonen. Automatic selection of context configurations for improved class-specific word representations. arXiv preprint arXiv:1608.05528, 2016.

[18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[20] Weijia Shi, Muhao Chen, Pei Zhou, and Kai-Wei Chang. Retrofitting contextualized word embeddings with paraphrases. arXiv preprint arXiv:1909.09700, 2019.

[21] Billy Chiu, Simon Baker, Martha Palmer, and Anna Korhonen. Enhancing biomedical word embeddings by retrofitting to verb clusters. In Proceedings of the 18th BioNLP Workshop and Shared Task, pages 125–134, 2019.

[22] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

[23] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649, 2018.

[24] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27, 2015.

[25] Sebastian Nagel. Common crawl news corpus, 2016.

[26] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[27] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847, 2018.

[28] Rishabh Misra. News category dataset, 06 2018.

[29] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.