DataScience-Polimi at SemEval-2022 Task 8: Stacking Language Models to Predict News Article Similarity

Marco Di Giovanni[*] [†] **Thomas Tasca**[*] [‡] **Marco Brambilla**[†]
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy
[*]These authors contributed equally to this work.
[†] {marco.digiovanni, marco.brambilla}@polimi.it [‡] thomas.tasca@mail.polimi.it

Abstract

In this paper, we describe the approach we designed to solve SemEval-2022 Task 8: Multilingual News Article Similarity. We collect and use exclusively textual features (title, description and body) of articles. Our best model is a stacking of 14 Transformer-based Language models fine-tuned on single or multiple fields, using data in the original language or translated to English. It placed fourth on the original leaderboard, sixth on the complete official one and fourth on the English-subset official one. We observe the data collection as our principal source of error due to a relevant fraction of missing or wrong fields.

1 Introduction

SemEval-2022 task 8 (Chen et al., 2022) (Multilingual News Article Similarity) is a document-level similarity task on news articles data. The goal is to predict whether two multilingual news articles cover the same real-world happening regardless of their writing style, political spin and tone. The task included resources written in 10 different languages: English, German, Spanish, Turkish, Polish, Arabic, French, Chinese, Italian and Russian (the training dataset included news written only in the first seven languages). This task is interesting as we can apply the obtained approaches to cluster news articles and track the similarity of news coverage between different outlets or regions as done in the Agenda Setting project[1].

Our best model is a simple but effective Stacking of a set of Language Models trained on different combinations of textual features. We fine-tune 14 Language Models, half of them with original multilingual texts and half with texts translated to English. We select three textual fields from the features extracted by our scraper (title, body and description of the news article), and we fine-tune a model for every combination.

Our model achieved a maximum Pearson correlation score of 0.790 and scored 4th on the leaderboard that considers the best test result for each team. However, the final ranking, based on a bootstrapping approach across teams' submissions to estimate the expected rank, penalizes us to 6th place since it assumes that submissions are an exploration of the hyperparameter/model configuration space of the system. The assumption, released after the end of the competition, does not hold for our team.

Our model mainly struggles with missing or wrong data since the training and evaluation datasets were released as links to scrape due to privacy policies. We noticed that we could not collect parts of the datasets, and some of the collected data were clearly wrong (e.g. "Get in touch with us. All rights reserved" as the body of an article).

The code will be available on GitHub[2].

2 Background

2.1 Task Setup

The organizers of the competitions provided two datasets: the training dataset and the test dataset. The training dataset is a collection of 4964 pairs of links to news articles with gold labels: real numbers ranging from 1 to 4, where 4 represents completely different articles. The test dataset is a collection of 4953 pairs of links to news articles without gold labels. Both datasets included the languages of the original articles so that we do not need to infer them from the data. Both datasets also included duplicated rows that we discarded. Due to copyright problems, the datasets did not directly contain the contents of the articles but only the links (original link and the Internet Archive version) to scrape them with a public script. Figure 1 shows the distribution of languages is in the two datasets. The test data contains news written in languages never

[1] http://www.euagendas.org/

[2] https://github.com/DataSciencePolimi/MultilingualNewsArticleSimilarity

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1229 - 1234

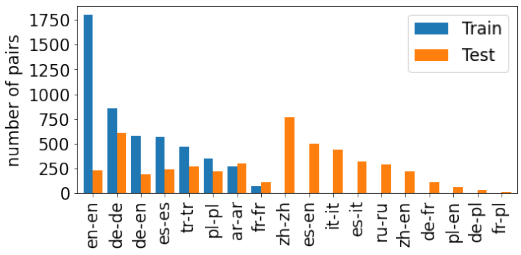

Figure 1: Distribution of language pairs in both datasets.

used in the training dataset (Chinese, Italian and Russian). Moreover, only 577 training pairs of news articles are in different languages (English and German). The performance of the models was computed using the Pearson Correlation Coefficient.

2.2 Related Work

We relied on transformers-based models and, in particular, we leverage the SentenceTransformers (Reimers and Gurevych, 2019) framework, which learns meaningful sentence Embeddings using Transformer-based Language Models (Vaswani et al., 2017). We build our solutions on top of the best pre-trained models provided and suggested by SentenceTransformers. Both models are based on Microsoft MPNet (Song et al., 2020), a pre-training approach that inherits the advantages of BERT's Masked Language Modeling (Devlin et al., 2019) and XLNet's Permuted Language Modeling (Yang et al., 2020) and avoids their limitations, providing better performances.

3 System overview

3.1 Data retrieval

To facilitate re-hydrating the textual content of the news articles, organizers provided a script[3] that downloads the earliest available version of each one of them from the Internet Archive and, only in case of problems, attempt to download them from the original site of publication using newspaper3k. For each article, we obtain the HTML content of the page and a JSON file containing additional information extracted from the page. We select the article's title, body and a brief description as features to feed our LMs.

In populating the dataset, we encountered two main challenges:

- We could not download the complete training dataset due to sites inaccessibility issues or anti-scraping systems. We did not encounter this issue on news articles from the Test dataset;

- The JSON files contains missing and noisy data. We believe that empty, unusable, or obviously incorrect fields are due to the low robustness of newspaper3k when applied to non-standard news websites.

While the first issue is hard to solve *a-posteriori* and could have been tackled by downloading the data as soon as the links were released, we improved the quality of the obtained dataset with Trafilatura[4] (Barbaresi, 2021), an alternative to newspaper3k. Trafilatura is an accurate web scraping tool for text discovery and retrieval that allows to prioritize the precision of the collection, i.e. yielding less but cleaner data.

3.2 Key algorithms

3.2.1 Single-field Language Models

As baseline models, we fine-tuned pre-trained Transformer-Encoder Language Model on a single field. We initialize the LMs with the pre-trained models suggested by SentenceTransformers (Reimers and Gurevych, 2019): a version of MPNet fine-tuned with self-supervised contrastive learning objective over a 1B sentences pairs obtained concatenating more than 20 datasets and trained using a self-supervised contrastive learning objective[5], and a similar multilingual alternative [6]. Every model was downloaded from Huggingface (Wolf et al., 2020).

The selected models are trained to generate, from variable-length input texts, fixed-size dense embeddings that encode semantic similarity: similar documents are mapped to vectors close to each other. We minimize the MSE loss between the cosine similarity of the embeddings obtained from the LMs and the rescaled labels (details about how and why we rescale the original labels in Section 5.1).

We trained the Single-field LMs on the three selected fields independently: Title (T), Description

[3] https://github.com/euagendas/semeval_8_2022_ia_downloader

[4] https://github.com/adbar/trafilatura

[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[6] https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2

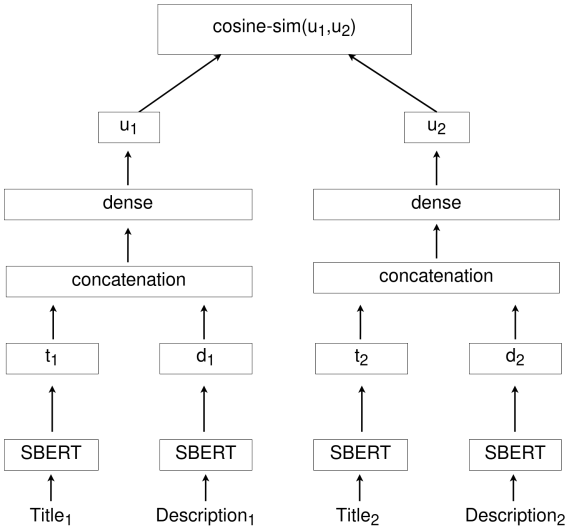

Figure 2: Schema of the two-fields (*title* and *description*) Language Model

(D) and Body (B). We obtain 6 different Single-field LMs, three of them fine-tuned on the original multilingual fields and three on the fields translated to English.

3.2.2 Multiple-fields Language Models

To improve the Single-Field LMs, we design an architecture that can handle multiple textual fields to generate semantically informative dense embeddings. Figure 2 shows the schema of such a model. Firstly, we select two fields (the alternative with the three fields is straightforward), and we feed them into a pre-trained Sentence-LM to generate two dense fixed-size embeddings for each news article (t_i and d_i). Then we concatenate the obtained embeddings, and we feed the result to a single learnable dense layer, initialized with Xavier (Glorot and Bengio, 2010), to generate a 768-dimensional global vector that includes information of both fields (u_i).

We train the model using every combination of two fields (TD, TB and DB) and also using all of them (TDB). We obtain 8 fine-tuned models, four of them fine-tuned on the original multilingual fields and four on the fields translated to English.

As for Single-field LMs, we minimize the MSE loss between the cosine similarity of the embeddings obtained from the LMs and the rescaled labels.

3.2.3 Stacking

Since ensemble learning has proven to be effective in competitions, we perform Stacking (Wolpert, 1992) on sets of the previously described models. We tested as final estimators simple regressors (implemented in scikit-learn (Pedregosa et al., 2011)) such as linear regressors regularized with Lasso (Tibshirani, 1996) or Ridge (Hoerl and Kennard, 2000), ElasticNet (Ela), Support Vector Machines (Cortes and Vapnik, 1995), KNN (Altman, 1992) and shallow Multi-layer Perceptrons (Hastie et al., 2001). Our best model is a MLP with 3 layers, 10 units per layer, trained with Adam (Kingma and Ba, 2015) and learning rate 3×10^{-2}, obtained using 5-fold cross-validation on the validation set.

4 Experimental setup

4.1 Data splitting, cleaning and preprocessing

We partitioned the available training data into two portions using an 80/20 training/test split stratified for language pairs and score (we approximate the labels to their integer part during this step).

We clean our dataset with the following preprocessing approaches:

- We remove duplicated rows. Some of the duplicates pairs obtained different similarity scores, so we replaced them with their mean;
- We detect the right character encoding with cChardet[7], a library to automatically detect the character encodings. Most of the files encoded with rare encodings were Turkish or Arabic news articles (encoded with legacy standards Windows-1254 and Windows-1256 respectively). When cChardet fails to detect the right encoding, we scrape the original website with Trafilatura;
- Some news articles had missing fields. We replace missing descriptions with bodies and missing titles with descriptions.

4.2 Translation

We fine-tuned our models on original data or translated data. Fine-tuning on original data requires a robust multi-lingual model that can compute semantic similarities regardless of the language of the documents. Fine-tuning on data translated to English requires an accurate Neural Machine Translation (NMT) model. We believe that Stacking both alternatives improves the overall performance since the final prediction relies on the strengths of both.

[7] https://github.com/PyYoshi/cChardet

We translated the whole dataset using EasyNMT[8] , a container of different NMT models. In particular, we used the Opus-MT (Tiedemann and Thottingal, 2020) model provided by the Helsinki NLP group.

4.3 Hyperparameters

Every LM was fine-tuned using the default optimizer (AdamW) with learning rate 2×10^{-5} and weight decay 0.01, using 10% of train iterations as warm-up. We trained the models for 3 epochs on our Training set, and we select the best model as the one that maximizes the Pearson Correlation Coefficient on the Validation set. Every experiment was performed using Colab (Bisong, 2019).

4.4 Evaluation measure

We evaluate our models using Pearson Correlation Coefficient of the cosine similarity between the embeddings generated by the models and the similarity manually scored by annotators. The scores ranges from 1 (perfect positive correlation) to -1 (perfect negative correlation), where 0 represents uncorrelated data.

5 Results and Ablation

Table 1 shows the results of our models. We obtain the best performing model on the Test set by Stacking 3 carefully selected Single-field Models fine-tuned respectively on translated titles, translated bodies and original bodies. The final estimator is a MLP with 3 layers and 10 units per layer as described above.

However, we obtained our best *submitted* prediction using our best performing model on the Validation set: a Stacking of all Single-field and Multiple-fields LMs, with the same final estimator as before. This model allowed us to score fourth in the original leaderboard that includes the best performing model for each team, and sixth on the official ranking[9].

[8] https://github.com/UKPLab/EasyNMT

[9] The final ranking was computed using a bootstrapping approach across teams' submissions. This approach assumes that multiple submissions by the same team represent an exploration of the hyper-parameter/model configuration space. This approach allows the organizers to estimate the general performance of the proposed models. However, the organizers announced this evaluation procedure after the end of the competition. We remark that the assumption does not hold for our team, and penalizes us on the final leaderboard. To get an idea of the overall test performance, several submissions from our team, especially at the beginning of the challenge, were generated with simple (sometimes even not fine-tuned) models. Moreover, due to technical problems of the challenge, we submitted opposite predictions from the same model to be sure that at least one scored correctly.

Name	Field	Dev	Test
MPNet	T_t	75.1	63.7
M_MPNet	T	74.7	65.1
MPNet	D_t	69.9	57.2
M_MPNet	D	67.6	56.9
MPNet	B_{t512}	81.7	77.6
M_MPNet	B_{512}	78.2	73.4
MPNet	T_t,D_t	77.4	66.7
M_MPNet	T,D	76.7	67.9
MPNet	T_t,B_t	81.5	75.8
M_MPNet	T,B	80.1	74.2
MPNet	D_t,B_t	80.9	74.5
M_MPNet	D,B	77.7	71.7
MPNet	T_t,D_t,B_t	81.9	75.1
M_MPNet	T, D, B	79.7	73.6
Ridge	B_{t512},T_t	82.13†	78.07
SVR	B_{t512},B_{512}, T_t	82.99†	78.71
MLP	A	**83.7**†	79.0
MLP*	B_{t512},B_{512}, T_t	83.5†	**79.2**
Winner model	/	/	*81.8*

Table 1: Pearson's $r \times 100$ on Validation and Test Datasets. We indicate with T the title, D the description and B the body of the articles. A stands for title, description and body, both original and translated, both obtained with Single-field and Multiple-fields approaches. The subscript t indicates translated texts. Values marked with † refer to the models evaluated with 5-fold crossvalidation on the validation dataset. All fields are truncated at 256 tokens except when specified with a subscript. The model marked with * was not submitted before the end of the competition.

The first part of Table 1 reports the results of the six Single-field LMs. As expected, the model fine-tuned on the body (B) of the news articles, the longest, thus most informative field, obtained the best results, both in the Validation and Test set. Models fine-tuned on the description (D) perform worse, probably due to the higher noise of the field (i.e., sometimes the content was missing or contained general information about the journal or the website). English models (reported as MPNet to highlight their initialization model) trained on translated data generally perform better than multilingual models (reported as M_MPNet) trained on original data.

The second part of the table reports the results of the six Double-fields LMs and the two Triple-

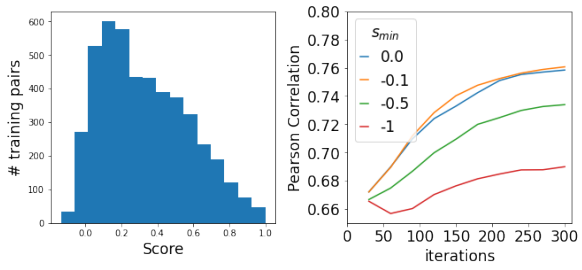

Figure 3: (*left*): Distribution of scores of pairs of titles from training set. (*right*): Pearson correlation during training for different values of s_{min}.

fields LMs. Combinations and interactions of fields give performances worse than our best performing Single-field LMs (MPNet on Body). We believe that this degradation of performance is due to two main factors. First, the noisiness of the title and description, as a higher percentage was missing or wrong. Second, due to memory issues, we had to perform Multiple-field LMs training reducing the maximum length of the body to 256 tokens instead of 512, as used when training Single-field LMs.

The third part of the table shows performances of our best four Stacking models. We evaluated many final estimators and combinations of first estimators and we submitted the scores from the models that performed better on our validation set. We computed the validation performance of stacking models using 5-fold cross-validation on the original validation dataset.

Finally we report the performance of the best team that participated to the competition. Up to now we do not know details about their approach.

5.1 Label rescaling

When we train our models we have to linearly scale the labels from the original range [1, 4]. While the straightforward choice of the final range could be [-1, 1] since our scores naturally fits that range due to the nature of the final cosine similarity computation, we observe that in practice, there are no pairs of titles from the training dataset that score less than -0.15 when we use a pre-trained model (see Figure 3 (left) for the complete distribution). Thus, we treat the lower bound of the transformed range s_{min} as an hyper-parameter to set. Figure 3 (right) shows values of Pearson Correlation on the validation dataset during the first training epoch for different choices of this parameter. We noticed that setting $s_{min} = -1$ as previously hypothesised leads to a slow training phase. We find $s_{min} = -0.1$ the best value among the tested ones. On the contrary, setting the higher bound $s_{max} = 1$ is optimal.

Language Pair	Test
ar-ar	66.2
de-pl	68.4
de-fr	71.3
pl-pl	71.5
ru-ru	72.9
zh-zh	76.8
es-it	77.3
de-de	78.0
tr-tr	78.3
de-en	82.2
it-it	82.4
zh-en	82.6
es-es	82.9
es-en	83.3
fr-fr	86.2
en-en	86.7
pl-en	87.1
fr-pl	88.3

Table 2: Pearson's $r \times 100$ of our best model on subsets of the Test dataset.

5.2 Error analysis

We report in Table 2 the performance of our best model for each language combination of the Test set. We believe that lower correlations are due to scraping issues, translation issues and to the different distribution of languages between the Training and Test sets.

6 Conclusion

To quantify the similarity between news articles, we propose an approach trained exclusively on the extracted textual data. We initialize our architecture with SOTA Semantic-Similarity Language Models, which we fine-tuned on titles, descriptions and texts of the articles. We also design a simple variant to process many textual fields at once. Finally, we perform stacking with a simple MLP, as it was proved to improve the overall performance of models trained on different features. Results show how the approach successfully estimated similarities since the main sources of error involved missing or wrong data.

References

N. S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Adrien Barbaresi. 2021. Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131. Association for Computational Linguistics.

Ekaba Bisong. 2019. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA.

Xi Chen, Ali Zeynali, Chico Q. Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw A. Grabowicz, Scott A. Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 Task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. In *Machine Learning*, pages 273–297.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Arthur E. Hoerl and Robert W. Kennard. 2000. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.