OversampledML at SemEval-2022 Task 8: When multilingual news similarity met Zero-shot approaches

Mayank Jobanputra * **Lorena Martín Rodríguez** *

Department of Linguistics, University of Tübingen, Tübingen, Germany
{mayank.jobanputra, lorena.martin-rodriguez} @uni-tuebingen.de

Abstract

We investigate the capabilities of pre-trained models without any fine-tuning, for a document-level multilingual news similarity task of SemEval-2022. We utilize title and news content with appropriate pre-processing techniques. Our system derives 14 different similarity features using a combination of pre-trained `MPNet` with well-known statistical methods (i.e. TF-IDF, Word Mover's distance). We formulate the multilingual news similarity task as a regression task and approximate the overall similarity between two news articles using these features. Our best performing system achieved a correlation score of 70.1% and was ranked 20^{th} among the 34 participating teams. In this paper, in addition to a system description, we also provide further analysis of our results and an ablation study highlighting the strengths and limitations of our features. We make our code publicly available at https://github.com/cicl-iscl/multinewssimilarity.

1 Introduction

Assessing semantic similarity between two given content pieces has become one of the important natural language processing (NLP) tasks. This task can help researchers estimate the quality of their models for many other tasks such as: machine translation (MT), summarization, question answering (QA), semantic search, dialog, and conversational systems. Extending semantic similarity task to a cross-lingual setup can extend the evaluation benefits to cross-lingual tasks as well. Previous works (Agirre et al., 2016; Cer et al., 2017) focus on sentence level cross-lingual semantic similarity. The presented shared task, Chen et al., 2022, proposes a novel problem that focuses on document-level semantic similarity based on news articles.

The multilingual news similarity task contains monolingual as well as cross-lingual pairs of news reports. This setup enables researchers to test their multilingual models on a document-level semantic similarity task. There can be multiple extensions to this, including the clustering of news articles and tracking similarity of news coverage between different outlets or regions.

In this work, we investigate the capabilities of pre-trained multilingual language models (LMs) as well as word-embedding models in this task, without fine-tuning them on the task data. Our solution pipeline combines pre-trained `MPNet` (Song et al., 2020) with well-known statistical methods with well-known statistical methods (i.e. TF-IDF, Word Mover's distance) to derive the semantic similarity features between articles using their *title* and *textual content*. We use these similarity features to approximate overall similarity between article pairs.

Our system performance is encouraging for this task, albeit with room for improvement. In the following sections, we describe our approach and provide a detailed study of the errors made by the system. We also report the results of an ablation study highlighting strengths and limitations of our derived features.

2 Task Setup

2.1 Dataset

The shared task introduced a new dataset consisting of 4964 article pairs in the training set and 4953 article pairs in the hidden test set. The participants were provided with these news articles' URLs and a Python script to scrape the texts. The training data contained an annotated overall similarity score for each pair and other similarity scores corresponding to features such as geography, entities, time, narrative, style, and tone.

The released training data consisted of monolingual pairs in English, German, Spanish, Turkish, Polish, Arabic, and French, and one cross-lingual

* Both authors contributed equally.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1171 - 1177

pair: German-English. The evaluation data contained 4,953 news article pairs. To the languages included in the training data, there were added monolingual pairs in Italian, Russian, and Chinese, and the cross-lingual pairs: German-French, German-Polish, Spanish-English, Spanish-Italian, French-Polish, Polish-English, and Chinese-English. For more details, please refer to table 5 and 6 in the Appendix. These monolingual and cross-lingual news article pairs are annotated on a 4-point scale from most to least similar.

2.2 Evaluation

The overall similarity between the two news stories is the only score used to evaluate system performance. The online scoring system calculates Pearson's correlation between system-generated overall similarity ratings and the gold standard ratings.

3 System Overview

In our approach, we formulate the multilingual news article similarity as a regression task, relying on different similarity features to approximate the overall similarity between two news reports. We choose this approach as the language pair distribution differs significantly between training set and the test set. This setup enables us to investigate the capabilities of pre-trained multilingual language models (LMs) as well as word-embedding models on document-level similarity task without fine-tuning.

We divide the news similarity task into a pipeline of five subtasks: Article Scraping, Preprocessing, Embedding Creation, Feature Calculation, and Inference. Figure 1 illustrates the architecture of our system.

3.1 Article Scraping

We used the script[1] provided by the organizers to download the article content from the web. After multiple tries, we were able to retrieve news content for 4940 out of 4964 training article pairs (see Table 5 in the Appendix). Similarly for evaluation data, we were able to retrieve news content for 4903 out of 4953 article pairs (see Table 6 in the Appendix).

3.2 Preprocessing

Our preprocessing step takes an article as an input and generates a json object containing only the information which is relevant for our approach. As we do not fine-tune any model based on the textual content, data cleaning is an important step for our system. We remove irrelevant content from the data in the following manner:

Copyright text: The article texts sometimes contains information regarding the copyright policy of the news websites. We observed a few cases where the scraper only downloaded the copyright notice instead of the news content. In such cases, copyright content increased or decreased the similarity significantly. To avoid errors in similarity feature calculation, we remove such copyright lines.

URLs: Generally, news reports also link other relevant references in their articles but parsing them to make them useful can be tricky. Moreover, sometimes these can contain unrelated advertisement links. Hence, we remove all kinds of links from the text for our purposes.

Cookies text: Similar to the copyright content, the scraper also ends up downloading text with information regarding the usage of cookies from the website. As this text is irrelevant for measuring content similarity, we remove this information from the article text.

Image captions: News stories can also contain images referring to some event or place that is covered in the news article. Such images are generally captioned with the details of the photographer credits. We find such credits irrelevant for our task and clean them from the article text.

3.3 Embedding Creation:

We use vector representation of article title and text to compute similarity features. We obtain these representations using `MPNet` and `FastText`. We calculate these vector representations once and store it on the disk. This way, we do not have to calculate these representations every time we need them. We utilize a multilingual version (Reimers and Gurevych, 2020) of `MPNet` (Song et al., 2020), fine-tuned on a paraphrasing task using parallel data for 50+ languages for calculating vector representations. We also experimented with language-agnostic BERT (Feng et al., 2020), but we dropped it after initial results.

For the longer text, multilingual `MPNet` model simply truncates the text and returns the representation of the first 512 tokens. We need a way

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

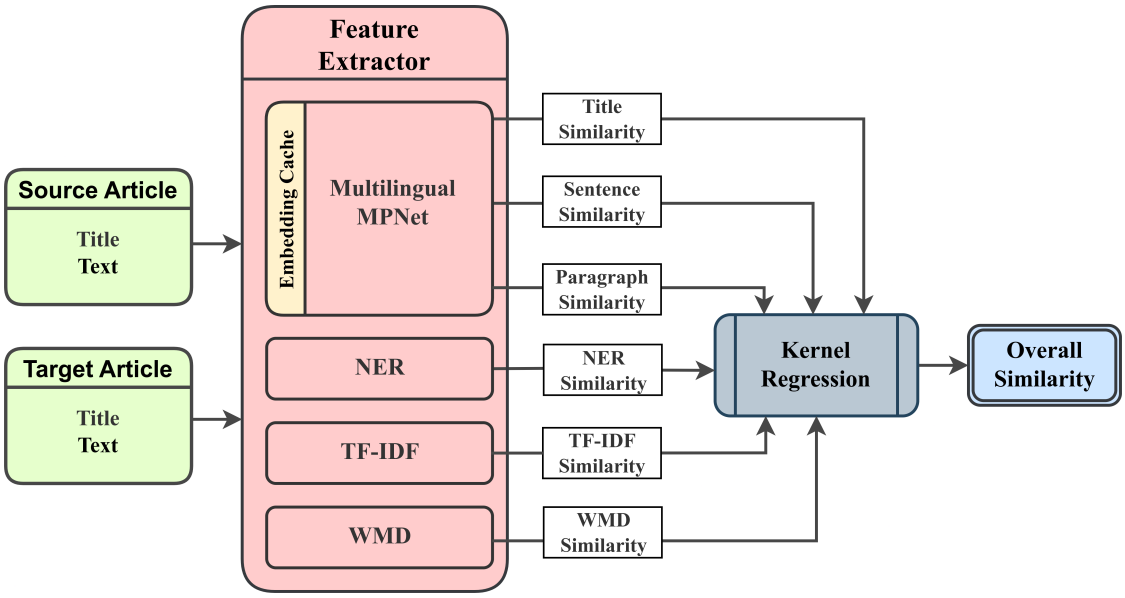

Figure 1: A schematic of our approach. We utilize Zero-shot feature extractors to derive similarity features from the articles. We use these features to train our regression model and obtain overall similarity score.

to be able to compute accurate vector representation of news articles irrespective of their length. Hence, we compute representations for article text at two different granularity levels: sentence-level and paragraph-level.

We obtain separate sentences using a multilingual sentence tokenizer from `SpaCy`.[2] For paragraph-level representation, we tokenize the text using a pre-trained tokenizer, provided along with `MPNet` model, and take the first and the last 512 tokens of the article. We obtain the vector representations of these sentences, the first 512 tokens, and the last 512 tokens using the `SentenceTransformers` (Reimers and Gurevych, 2019) library. Similarly, we compute the vector representations of the news title. From now on, we will refer to the first 512 tokens as the `first paragraph` and the last 512 tokens as the `last paragraph`. Note that for articles shorter than 512 tokens, the first and the last paragraph will be the same.

3.4 Feature Calculation

To estimate overall news article similarity, we derive 14 unique similarity features from the news title and text pairs. In this subsection, we describe all the features in detail.

[2]https://spacy.io/models/xx

3.4.1 Sentence similarity

We derive four similarity features from the article text to capture the sentence-level similarity between articles. These features are obtained as follows:

Sentence mean similarity: The average cosine similarity scores of the top matching sentence vectors between the source and target articles.

Sentence maximum similarity: The maximum cosine score of the top matching sentence vectors between the source and target articles.

Sentence minimum similarity: The minimum cosine score of the top matching sentence vectors between the source and target articles.

Sentence median similarity: The median of the cosine similarity scores of the top matching sentence vectors between the source and target articles.

3.4.2 Paragraph similarity

Additionally, we derive two similarity features from the article text to capture the paragraph-level similarity between articles. These features are obtained as follows:

First paragraph similarity: The cosine similarity value between the `first paragraph` vector representations of the news articles.

Last paragraph similarity: The cosine similarity between the `last paragraph` vector represen-

tations of the news articles.

3.4.3 Title similarity

The title should summarize the article text in a meaningful manner. We calculate five similarity features from the article title, capturing title similarity as well as inter and intra title-text similarity between articles. These features are obtained as follows:

Title similarity: The cosine similarity between the title vector representations of the news article pair.

Inter title-text similarity: We calculate inter title-text similarity to see how the source title is similar to the text of the target and vice versa. We utilize stored vector representations of the corresponding news entities and calculate cosine similarity between these entities. Note that we produce two separate features for inter title-text similarity (i.e. $sim(\text{title}_s, \text{text}_t), sim(\text{title}_t, \text{text}_s)$).

Intra title-text similarity: Similarly, to measure title-text coherence, we calculate intra title-text similarity between the title and the text of same news article using the stored vector representations. Note that we produce two separate features for intra title-text similarity (i.e. $sim(\text{title}_s, \text{text}_s), sim(\text{title}_t, \text{text}_t)$).

3.4.4 NER similarity

We calculate named entity similarity NE_{sim} using the below equation.

$$\text{NE}_{sim} = \frac{|\text{NE}_s \cap \text{NE}_t|}{max(|\text{NE}_s|, |\text{NE}_t|)}$$

where NE_s represents set of named entities in the source article, NE_t represents set of named entities in the target article.

3.4.5 TF-IDF Similarity

We first remove stop words using `NLTK`'s language-specific stop words corpus. We then estimate the cosine similarity between TF-IDF representations of the article pair.

3.4.6 WMD Similarity

Similar to TF-IDF similarity, we calculate the Word Mover's distance of two texts without stop words using multilingual `FastText`(Bojanowski et al., 2017) model from `Gensim`.[3]

[3]https://fasttext.cc/docs/en/crawl-vectors.html

3.5 Inference

The last part of our system is estimating the overall similarity using the features obtained. We experimented with three different setups: regression over all the features, multitask regression using additional available scores, and regression over reduced feature space (i.e. principle components, autoencoder representations). In this subsection, we mention all the setups briefly and provide details of our best performing system.

Multitask Regression: The training data contains similarity scores for geography, entities, time, narrative, style, and tone along with Overall article similarity. We trained a Multi-task Lasso model and Multi-task autoencoder on the training set but after initial experiments, we decided not to pursue these setups further.

Regression over reduced feature space: We apply principal component analysis on our feature space for dimensionality reduction.

Regression over entire feature space: We experimented with Linear regression, Decision Tree regression, Random Forest regression, Kernel ridge regression, Multilayer perceptron regression, and TabNet regression (Arık and Pfister, 2021).

Our two best performing systems used Kernel ridge regression. In the highest ranked system, we train a Kernel ridge regressor using a polynomial kernel of degree 3 and a regularization co-efficient of 1.0. This way we allow our model to learn from non-linear features. Our second best performing system uses Kernel ridge regression (KRR) with RBF kernel over top-4 principal components and achieves 70% Pearson correlation. TabNet regression ranked third during the evaluation phase, with a correlation score of 69.1%. We describe the implementation details of our best performing setup below.

Implementation details: We split the data into 95:05 training and evaluation datasets and utilize the entire feature space. Going from 80:20 split to 95:05 split boosted our model accuracy by half point. As a final step, we also clip the model predictions between 1-4 to make sure that our model predictions remain inside the range. We use `scikit-learn` (Pedregosa et al., 2011) for training the kernel regression model and `wandb` (Biewald, 2020) for hyperparameter tuning.

4 Results

All the submissions for this shared task were evaluated with regard to Pearson's correlation coefficient. Our best performing system achieved a score of 70.1% in the evaluation phase. We were officially ranked 20^{th} out of 34 teams on the main task leaderboard. On the English-only subtask, we were ranked 15^{th} out of 34. We report scores for our top three submissions during evaluation phase in Table 1. All three systems use all the features to predict the overall similarity.

Model	Data Split	Score
KRR-poly	95:05	70.1
KRR-rbf	80:20	70.0
TabNet	95:05	69.1

Table 1: Correlation scores of our top performing systems on the hidden test set

We report our results for each monolingual language pairs in Table 2 and cross-lingual pairs in Table 3. The general system performance was similar for monolingual and cross-lingual pairs, with an average accuracy of 0.71 for monolingual pairs and 0.72 for cross-lingual ones.

Language	Score	Language	Score
fr-fr	84.34	tr-tr	70.36
es-es	81.24	zh-zh	64.42
en-en	79.55	ar-ar	62.23
it-it	79.42	pl-pl	61.49
ru-ru	73.50	de-de	59.43

Table 2: Correlation scores for monolingual pairs

Language	Score	Language	Score
pl-en	82.84	fr-pl	74.76
es-en	80.13	es-it	70.95
zh-en	76.91	de-pl	60.11
de-en	76.54	de-fr	55.47

Table 3: Correlation scores for multilingual pairs

The highest accuracy was achieved in the French monolingual pair, with 0.84, but the lowest accuracy score was found in the German-French cross-lingual subsection with 0.55. This can be explained by the general low results that the system achieved in the pairs with news written in German, be it in the monoligual subset (0.59), or cross-lingual pairs German-French and German-Polish (0.60). Only one pair with German language had an above average accuracy, the German-English cross-lingual pair (0.76).

5 Performance Analysis

In this section, we analyze the performance of feature sets used by our system and compare different subsets against each other. We also discuss some errors made by our system and possible improvements.

5.1 Ablation study

We conducted ablation experiments to evaluate the importance of different feature sets on the results. We use released test set labels and our best performing model, Kernel Regression to conduct this study. We report the results of this study in Table 4. Note that these results were obtained after extensive hyperparameter tuning, which was not feasible during the shared task evaluation phase.

Features	Correlation
MPNet features	71.83
Non-MPNet features	47.15
MPNet features + WMD distance	71.35
TF-IDF + WMD distance	47.89
MPNet Sentence features	63.21
MPNet Title features	64.11
MPNet Paragraph feature	63.73

Table 4: Results of feature ablation study

We observe that features derived from `MPNet` perform better than our reported system and slightly improve the correlation score. We also observe that adding other features with `MPNet`-derived features rather degrades the system performance (i.e. `MPNet` features + WMD distance). Other features perform very poorly compared to `MPNet` features through our setup. While `MPNet` feature sets perform substantially better individ-

ually compared to other features, combining all MPNet gives the best performance on the task.

5.2 Error analysis

In order to closely examine the performance of our system, we analysed a subset of the news pairs which were classified in the wrong category. This subset only included news pairs written in English, German, Spanish, Italian or French and its cross-lingual combinations. Most differences between our systems' result and the annotation guidelines differed in less than one point. Since the different categories of similarity proposed in the dataset also differed in one point, we considered this as our threshold for error analysis.

The following patterns were found when our system overestimated the similarity between two news pairs:

Articles with parallel structures: Some news genres present a more fixed structure than others. Such is the case for police reports, which include similar key phrases but narrate different events.

Same location: Two different events which happened in the same location.

The following patterns were found for when our system underestimated the similarity between two news pairs:

Scraping errors: At least one of the articles did not contain any textual content relevant for the news article.

Lack of information: At least one of the news articles was extremely brief (less than one paragraph), and lacked information about the event.

Different titles: The titles focused on different aspects of the news report.

6 Conclusion

In this paper, we have described our participation in the Task 8 of SemEval-2022, "Multilingual news article similarity". We developed a system to investigate the capabilities of pre-trained language models (LMs) as well as word-embedding models. Our results suggest that the system performs similarly for monolingual and cross-lingual pairs, but its performance varies based on the specific language pairs. The ablation study showcases the strength of the pre-trained multilingual language models for this task. Given the performance of our system, despite the noticeable variation in language-pair distributions, we speculate that our approach can be used to deliver similar results for additional languages as well.

For the future, we would like to explore the system's performance with the same features but also fine-tuning our MPNet model on the training dataset. This would allow us to compare the effectiveness of pre-trained models against the fine-tuned model. Additionally, we would like to experiment with new features such as article summary similarity and article topic similarities.

Acknowledgements

We would like to thank Prof. Çağrı Çöltekin for the his inputs and helpful discussions.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Sercan O Arık and Tomas Pfister. 2021. Tabnet: Attentive Interpretable Tabular Learning. In *AAAI*, volume 35, pages 6679–6687.

Lukas Biewald. 2020. Experiment Tracking with Weights and Biases. Software available from wandb.com.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 Task 1: Semantic Textual Similarity, Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

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.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT Sentence Embedding. *arXiv preprint arXiv:2007.01852*.

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. *the Journal of machine Learning research*, 12:2825–2830.

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*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 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. *Advances in Neural Information Processing Systems*, 33:16857–16867.

A Appendix

Language	**#pairs (w\o dup)**	**Downloaded pairs**
en-en:	1800	1773
de-de:	857	853
de-en:	577 (575)	522
es-es:	570	561
tr-tr:	465	428
pl-pl:	349	349
ar-ar:	274	274
fr-fr	72	72

Table 5: Distribution of languages in the training data

Language	**#pairs (w\o dup)**	**Downloaded pairs**
en-en	236	236
de-de	611 (608)	608
de-en	190 (185)	185
es-es	243	243
tr-tr	275	272
pl-pl	224	224
ar-ar	298	298
fr-fr	111	111
zh-zh	769	764
es-en	498 (496)	496
it-it	442 (411)	411
es-it	320	310
ru-ru	287	287
zh-en	223 (213)	213
de-fr	116	111
pl-en	64	64
de-pl	35	34
fr-pl	11	11

Table 6: Distribution of languages in the test data