SkoltechNLP at SemEval-2022 Task 8: Multilingual News Article Similarity via Exploration of News Texts to Vector Representations

Mikhail Kuimov **Daryna Dementieva** **Alexander Panchenko**
Skolkovo Institute of Science and Technology
{Mikhail.Kuimov, Daryna.Dementieva, A.Panchenko}@skoltech.ru

Abstract

This paper describes our contribution to SemEval 2022 Task 8 on Multilingual News Article Similarity. The aim was to test completely different approaches and distinguish the best performing. That is why we've considered systems based on Transformer-based encoders, NER-based, and NLI-based methods (and their combination with SVO dependency triplets representation). The results prove that Transformer models produce the best scores. However, there is space for research and approaches that give not yet comparable but more interpretable results.

1 Introduction

The SemEval 2022 Task 8 competition (Chen et al., 2022) aims to develop systems that identify multilingual news articles that provide similar information. This is a document-level similarity task in the applied domain of news articles, rating them pairwise on a 4-point scale from most to least similar. The alikeness of news is measured in such a sense: how similar are them in geography, time, shared entities, and shared narratives. The developed approaches for solving the task can be applied to several different real-world tasks. The first one is the clustering of news.

A lot of news-providing companies want thousands of articles from different publishers on one topic to be combined in a single page showing the news.

Another task is Fake News Detection. This task has become extremely important to solve lately. Different articles on the same news story can be compared to find contradictions in facts and details. This can be an indicator that the story is fake like it is shown in (Dementieva and Panchenko, 2021).

2 Related Work

In (Montalvo et al., 2007), authors extract `PERSON`, `ORGANIZATION` and `LOCATION` named entities (NE) and compose a vector for each of the category with the help of Levenshtein distance function and TF-IDF weighting function, which combines Term Frequency (TF) and Inverse Document Frequency (IDF). These 3 vectors are compared to 3 corresponding vectors of the second news with cosine distance. Obtained scores are combined with the set of IF-THEN rules. In (Rahimi et al., 2019) the approach for cross-lingual transfer is proposed which is evaluated on the Named Entity Recognition task. Dialogue competition (Gusev and Smurov, 2021) on Russian news clustering has produced many promising methods which could be adopted to the multilingual case. Most of them, like (Sergei et al., 2021; Glazkova, 2021), are the variations of fine-tuning the transformer models and making ensembles.

In (Martín et al., 2021), the authors developed the pipeline for checking the news on veracity. They compare embeddings of the news under consideration with ones from the database, using cosine distance. Then they take the most similar news found and apply Natural Language Inference (NLI) model to obtain the probability that two texts contradict each other. This probability is used to decide whether the input news is fake. However, NLI scores can be used to find the similarity between articles.

3 Methodology

To solve the task we have tried several approaches. In the subsection 3.1 we will give an overview on methods exploiting pre-trained transformer models as the foundation. In the next subsection we will explain how the Natural Language Inference (NLI) problem can be reduced to the task of News Article Similarity. Block 3.3 is dedicated to approaches based on the Named Entities extracted from the news texts. In the last section, the approaches which were tested to improve the quality of prediction during the post-evaluation period will

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1136 - 1144
July 14-15, 2022 ©2022 Association for Computational Linguistics

be described.

3.1 Transformer-based Pre-trained Encoders

Pre-trained neural masked language models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) have shown superior performance on a wide range of NLP tasks both in monolingual and multilingual settings. During the work on the task of the competition, the approach for fine-tuning Transformers was developed. The following multilingual models were tested: DistilBERT[1], BERT[2] RoBERTa[3], XLM[4]. All these models support all the languages included in the competition dataset. Two different architectures were chosen for fine-tuning the language models. The first one is based on the approach for the BERT Next Sentence Prediction problem described in the original article (Devlin et al., 2019). We will call it **TransformerEncoder-CLS**. The second approach is inspired by the articles (Reimers and Gurevych, 2019; Sergei et al., 2021). It will be labeled as **TransformerEncoder-CosSim** from now on.

3.1.1 TransformerEncoderCLS

The general scheme of the approach is shown in Fig. 1. The Transformer model takes as input 2 tokenized news texts separated by [SEP] token, which is needed for the model to distinguish words from different texts. Also, this sequence of tokens has a special [CLS] token in the beginning. Passing through the layers of the model, each token results in the embedding vector. All the information from the sequence is aggregated in the [CLS] token embedding. That is why we use it as the input to the regression head, which is the combination of fully-connected layer and Sigmoid nonlinearity. The linear layer dimensions are $emb\_len \times 2$, where emb_len is the dimension of the hidden layer. We use the output probability of the first class as the similarity score. Togeth er with mapped to $[0, 1]$ range ground true similarity scores, the predicted scores are passed to the MSE loss function. Transformers weights are not frozen while training and initialized from the aforementioned pre-trained multilingual models. The models were trained on GPU NVIDIA GeForce RTX 3090 for 10 epochs with a learning rate of 10^{-5} and batch size equal to 8.

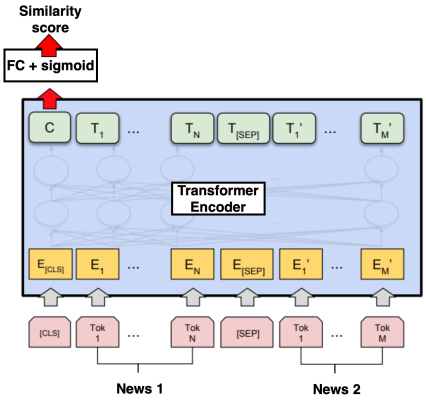


Figure 1: TransformersEncoderCLS architecture, depicted from the original paper (Devlin et al., 2019).

3.1.2 TransformerEncoderCosSim

The general scheme of the approach is shown in Fig. 2. The pre-trained Transformer model takes as input the tokenized news text. Then, Transformer output embeddings are passed through the average pooling followed by a fully-connected layer[5] and L2 normalization layer. This procedure is applied for both compared news. Then, the resulting text embeddings are passed in the cosine distance function which is computed with equation bellow to produce a distance score:

$$cosine\_dist = 1 - |cosine\_sim|.$$

We use absolute value of cosine similarity function because it takes values from -1 to 1. Together with mapped to $[0, 1]$ range ground true scores, the predicted scores are passed to MSE loss function. Transformers weights are not frozen while training and initialized from the aforementioned pre-trained multilingual models. The models were trained on GPU NVIDIA GeForce RTX 3090 for 10 epochs with a learning rate of 10^{-6} and batch size equal to 4.

3.2 Natural Language Inference

The task of estimation similarity between news contents can be reformulated as Natural Language Inference task, which is the main hypothesis tested in

[1] https://huggingface.co/distilbert-base-multilingual-cased

[2] https://huggingface.co/bert-base-multilingual-cased and https://huggingface.co/bert-base-multilingual-uncased

[3] https://huggingface.co/xlm-roberta-base and https://huggingface.co/xlm-roberta-large

[4] https://huggingface.co/xlm-mlm-17-1280

[5] The linear layer dimensions are $emb\_len \times emb\_len$, where emb_len is the dimension of the hidden layer

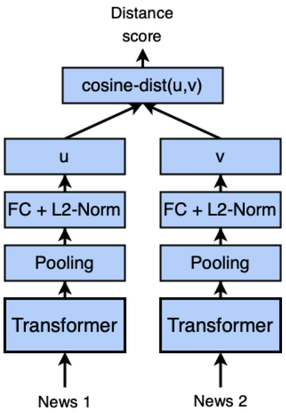
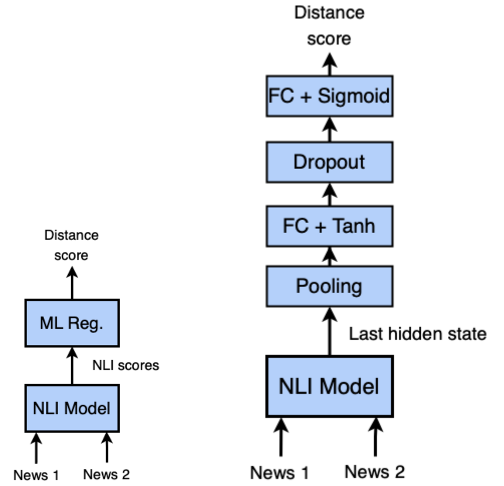


Figure 2: TransformerEncoderCosSim architecture.

this work. *Natural Language Inference* (NLI) is the problem of determining whether a natural language hypothesis h can reasonably be inferred from a natural language premise p (MacCartney and Manning, 2008). The relations between hypothesis and premise can be *entailment*, *contradiction*, and *neutral*. The release of the large NLI dataset (Bowman et al., 2015) and later multilingual XNLI dataset (Conneau et al., 2018) made possible the development of different deep learning systems to solve this task. That is why the pre-trained NLI models like XLM-RoBERTa-large appeared. We use this model pre-trained on multilingual XNLI dataset[6] to obtain NLI scores for pairs "the first news as premise $p \leftrightarrow$ the second one as hypothesis h". The size N of the used content is a hyperparameter of this NLI based approach for the news content similarity computation. NLI model outputs the probabilities of news pair to be classified as entailment, contradiction, or neutral. Hence, it's 3 real numbers from the $[0, 1]$ range. These extracted NLI features are passed as input to the Machine Learning model, which predicts the similarity score for the pair of news under consideration. In our work, we've compared the performance of several regression models: Linear Regression, Support Vector Machine for regression, Decision Trees, Random Forest, Gradient Boosting. The last one gave the best results. The general scheme of the approach is shown in Fig. 3. Also, several improvements to this pipeline were tested:

1. **Both pairs**. Each piece of news is used as a premise and hypothesis. As a result, we get twice more features for training.



(a) Basic architecture.

(b) Fine-tuning architecture.

Figure 3: NLI approach

2. **Subject-Verb-Object triplets**. We extract syntactic dependencies from the sentences of a text to make triplets consisting of subjects, verbs, and objects. These triplets are passed to the model. Such an approach shortens the input data, which makes the process of extracting NLI features faster and doesn't have the significant influence in quality of the method. We extracted syntactic dependencies with Spacy library (Honnibal et al., 2020).

3. **Fine-tune**. We fine-tune the NLI model on the data of the competition. The approach is based on the one proposed by (Martín et al., 2021). We add the regression head to the NLI model, which has global average pooling of the last hidden state of the transformer model, linear layer with 768 neurons and $\tanh$ activation, a 10% dropout for training, and a classifier linear layer with sigmoid. The output probability is treated as a similarity score, and MSE loss is used. This regression head is trained, freezing the XLM-RoBERTa-large weights to preserve the previous pre-training. This is optimized using Adam optimizer (Kingma and Ba, 2015) with 10^{-3} learning rate. The general scheme of the approach is shown in Fig. 3.

3.3 Named Entity Recognition

Transformers have great performance but almost no interpretability. In search of interpretability,

[6] https://huggingface.co/joeddav/xlm-roberta-large-xnli

the Named Entity Recognition-based approach has been developed. The general scheme of the approach is shown in Fig. 4. News texts are preprocessed and forwarded to the NER extractor to extract locations (LOC), organizations (ORG), and person entities (PER). For this task we've tested and compared several tools:

1. **Transformer for named entities tagging.** We used BERT[7] pre-trained model from the Hugging Face repository. It is a Named Entity Recognition model for 10 high-resource languages (Arabic, German, English, Spanish, French, Italian, Latvian, Dutch, Portuguese, and Chinese) based on a fine-tuned mBERT base model.

2. **Polyglot for Named Entity Extraction.** The models from this package (Al-Rfou et al., 2015) were trained on datasets extracted automatically from Wikipedia. Polyglot currently supports 40 major languages, including all presented in the dataset of the competition.

3. **Spacy.** Spacy library (Honnibal et al., 2020) provides huge variety of NLP tools, including NER extractor. We used multi-language model,[8] trained on Wikipedia.

In the next step, we vectorize extracted entities with Bag of Words, Tf-Idf, Fasttext (Bojanowski et al., 2017), Bert embeddings[9] for comparison. Then we average all the word vectors. As a result, we obtain 3 vectors (one for each of LOC, PER, ORG entities) for each text. Corresponding vectors for LOC, ORG, PER for two texts are compared with cosine distance to get 3 distance scores for every pair of news under consideration. Then, these scores are passed in the Machine Learning model to get the final distance score. We test several regression models: Linear Regression, Support Vector Machine for regression, Decision Trees, Random Forest, Gradient Boosting.

3.4 Additional study

To improve the quality of the prediction the following two techniques were tested:

1. **Augmentation.** Testing part of the dataset has a lot of language pairs[10] which are not presented in the training part of the dataset. To test the influence of unseen language pairs on the results, we added pairs of news for the missing language pairs. Such augmentation was performed with the help of the Google Translator, which was accessed with the help of Deep Translator python library. The pairs of news were selected randomly from the pairs written in English and then translated to the target languages. Samples were added to the training part of dataset in the same proportion they are presented in the testing part of the dataset. As a result, training dataset was extended to 7505 samples.

2. **Stacking.** Ensembling different models is a common way to improve the scores. To aggregate the dependencies caught by several models, we exploited the technique called stacking. To form the ensemble, we used TransformerEncoderCosSim, TransformerEncoderCLS, fine-tuned NLI model and NER model[11] which has shown the best results in the experiments described bellow. All the models were trained on three quarters of the training dataset. And one quarter of the dataset was used to train the aggregation model. We used Linear Regression model with L_2 regularization as aggregation model.

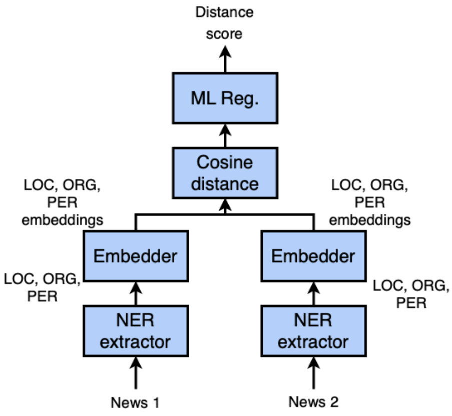


Figure 4: NER approach architecture.

4 Results

Our team has taken the 14th place among 32 participants. The best result for separate model, 0, 734 correlation, was reached by the TransformerEn-

[7] https://huggingface.co/Davlan/bert-base-multilingual-cased-ner-hrl

[8] xx_ent_wiki_sm

[9] bert-base-multilingual-uncased pre-trained model was used

[10] A pair of languages, in which a pair of news is written.

[11] We used the following combination: Huggingface NER tagger, Huggingface embeddings, Gradient Boosting ML model.

coderCosSim model. Final results for all separate methods, as well as the best competition score, are provided in Table 1. Also, we provide the results for ensembles of models in the Table 3. The application of ensembling and augmentation techniques improved the best result to $0,763$ correlation. In addition to the test set, which was provided by organisers during the evaluation period, the performance of the developed systems was evaluated on the validation set. Validation set was randomly sampled from the training data[12] in case of TransformerEncoder methods, including fine-tuned NLI model. For other methods the results on validation are the results obtained with 5-fold cross-validation.

	Validation	Evaluation
TransformerEncoderCLS	**0.813**	0.706
TransformerEncoderCosSim	0.793	**0.734**
NLI	0.478	0.477
NLI fine-tuned	0.670	0.632
NER	0.496	0.395
NLI + NER	0.615	0.546
Best SemEval result	—	0.818

Table 1: Overall results. Pearson correlation.

Transformer models. As it has already been said TransformerEncoderCosSim model has shown the best result. It was the one with XLM[13] pre-trained model. The worst score was given by the DistilBert model. We provide the comparison of different encoders from Transformers for 2 proposed models in the Table 6 in the appendix. As for the TransformerEncoderCLS model, its performance has dropped by 12% on the evaluation part of the dataset in comparison to validation part. And it's become worse than the TransformerEncoderCosSim model, although it showed better results on the cross-validation.[14] In general, the transformer-based models have a lower correlation on the evaluation data. You can see a similar behavior for the NLI fine-tuning approach.

NLI. The comparison of the results for NLI-based models is provided in Table 2. The best score for the NLI approach was given by the Gradient Boosting model. (We provide the comparison of results for different Machine Learning models for «NLI pairs - titles» in the Table 7 in appendix.) The fine-tuning approach has given the best correlation here. Also, there is a tendency for smaller input text to have better scores. The highest correlation was achieved when only titles were given as input. The reason for that could be that the NLI model was trained on the XNLI dataset, composed of short phrases. That is why it was decided to try to shorten the news with the extraction of SVO triplets from them. The extracted triplets were joined to form a text which was forwarded to the input of the NLI model. As you can see from Table 2 the quality of both methods (with fine-tuning and without) has dropped significantly. Hence, the conclusion is that despite SVO triplets give a good summary of the given text, they are not applicable, at least without any complex processing, for the task of comparing the news. Also, it could mean that the source of similarity of articles is not contained in Subjects, Verbs, and Objects. Last, it is worth mentioning that the resulting summary for big texts still has quite a large size in comparison to titles.
The idea to extract NLI scores from both pairs, as it was described in devoted subsection, gave an improvement. Also, it can be noticed that the NLI approach without fine-tuning is quite robust to adding new languages. The score for "NLI pairs - titles" has only a slight decrease on the evaluation dataset. Although the correlation for single NLI features is low, it becomes significantly better in combination with features with the NER-based method. This approach is described in more details in the devoted paragraph bellow.

	Validation	Evaluation
NLI tiltes	0.453	0.438
NLI pairs - titles	0.478	0.477
NLI pairs - titles + text	0.354	0.310
NLI pairs - SVO	0.154	0.107
NLI fine-tuned - titles	**0.670**	**0.632**
NLI fine-tuned - titles + text	0.627	0.589
NLI fine-tuned - SVO	0.495	0.422

Table 2: Comparison of NLI approaches. Pearson correlation.

NER. One can find the comprehensive comparison of different NER taggers, various vectorizing techniques and different Machine Learning models for prediction of distance score in the Table 5 in the appendix. You can see that the best correlation was shown by combination: Huggingface NER tagger, Huggingface embeddings, Gradient

[12] The size of validation set was 0.25 from the size of the training dataset.

[13] https://huggingface.co/xlm-mlm-17-1280

[14] Model which has shown the best result: https://huggingface.co/xlm-roberta-large

Boosting ML model. In general, Gradient Boosting has shown superior scores for all combinations of NER taggers and vectorizers. Also, Huggingface embeddings in combination with this model have shown the highest results for all vectorizing methods listed in the Methodology section. However, in comparison to NLI and Transformers approaches, the results for NER models are significantly lower. Looking at the outputs of the model (You can find the examples in Table 4 in appendix), the following behaviors can be noticed. In our method in cases when no named entities were found for the `PER`, `ORG` or `LOC` classes, the distance score was set to 0.5, because it is not clear whether the absence of named entities is an indicator of similarity or not. These 0.5 scores confuse the model, increasing its generalization error. The second problem is that when there is no overlap of named entities in one of the classes, it could lead to two bad outcomes. When the other two distance scores correctly reflect the ground true similarity, like in the second example in Table 4, the one with no overlap could be large, which spoils the overall prediction. The second behavior happens when the extracted entities have no straight overlap but happen to be similar in vector space. For example, two different news about the close locations. In this case, the model can output a small distance, which is not correct. Also, the errors of the NER tagger makes the model performance worse. As a result, the model tends to predict values from the middle of the $[1, 4]$ range, avoiding its edges. In addition, the problems described make the results even worse on unseen evaluation data. We provide the comparison of the best results for different NER extractors for validation and evaluation in the Table 8.

NER + NLI. As you can conclude from Table 1, NER features, having poor single performance, add significant improvement in correlation being combined with NLI features. To obtain this result we have taken the features used in best-scored NLI and NER models. For classification Gradient Boosting ML model was used as it had given the highest results for both approaches.

Additional study. The application of augmentation to the training part of the dataset improved the result of the best performing model from 0.734 to 0.746, which is a slight improvement. It can be concluded that the performance of this model is not highly effected by unseen language pairs. The increase in score may be caused just by the increase of the number of training samples.

	Correlation
TrEncCLS, TrEncCosSim	0.752
TrEncCLS, TrEncCosSim, NLI	**0.763**
TrEncCLS, TrEncCosSim, NLI, NER	**0.763**

Table 3: Comparison of the results for different ensembles on the evaluation dataset. Pearson correlation. The names of TransformerEncoders models were shortened.

The results for stacking of the models can be found in the Table 3. In this experiment stacking technique was combined with augmentation, which showed a slight improvement in score. You can see that the addition of the predictions obtained with the NER model gives no increase in score. Overall, the augmentation together with stacking gave the 4% improvement to the result of TransformerEncoderCosSim model.

5 Conclusion

We have tested several approaches, including two systems based on Transformer-based encoders, two NLI approaches (with fine-tuning and without), NER-based pipeline and the ensemble of these models. The best result was achieved by the ensemble of TransformerEncoderCosSim, TransformerEncoderCLS and fine-tuned NLI models. To improve the scores the following things can be done. For models based on Transformer-based encoders, sentence Transformers can be tested. To improve the NER-based method additional features can be added (addition of NLI features improved the correlation), and also we can apply the binary mask for the feature matrix not to take into account 0.5 values while calculating the loss during the training process can be applied.

Source code of our solutions is available online[15]. Also, the hyper-parameters of the models can be found there.

Acknowledgements

This work was supported by MTS-Skoltech laboratory on AI.

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. POLYGLOT-NER: massive

[15] `https://github.com/skoltech-nlp/multilingual_news_similarity`

multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, BC, Canada, April 30 - May 2, 2015*, pages 586–594. SIAM.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The 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.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.

Daryna Dementieva and Alexander Panchenko. 2021. Cross-lingual evidence improves monolingual fake news detection. In *Proceedings of the ACL-IJCNLP 2021 Student Research Workshop, ACL 2021, Online, JUli 5-10, 2021*, pages 310–320. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

A. Glazkova. 2021. Towards news aggregation in russian: a bert-based approach to news article similarity detection. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2021". Moscow, Russia (Online)*.

Ilya Gusev and Ivan Smurov. 2021. Russian news clustering and headline selection shared task. *CoRR*, abs/2105.00981.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

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*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, pages 521–528.

Alejandro Martín, Javier Huertas-Tato, Álvaro Huertas-García, Guillermo Villar-Rodríguez, and David Camacho. 2021. Facter-check: Semi-automated fact-checking through semantic similarity and natural language inference. *CoRR*, abs/2110.14532.

Soto Montalvo, Raquel Martínez-Unanue, Arantza Casillas, and Víctor Fresno. 2007. Bilingual news clustering using named entities and fuzzy similarity. In *Text, Speech and Dialogue, 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3-7, 2007, Proceedings*, volume 4629 of *Lecture Notes in Computer Science*, pages 107–114. Springer.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 151–164. Association for Computational Linguistics.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Khaustov Sergei, Kabaev Andrey, Gorlova Nadezda, and Kalmykov Andrey. 2021. Bert for russian news clustering. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue 2021". Moscow, Russia (Online)*.

Appendix

pair_id	NER 1	NER 2	dist. LOC	dist. PER	dist. ORG	Predict.	Ground true
1484012638 1483801741	**LOC**: Baku, Azerbaijan, Shamakhi, Ismayilli, Aghsu **PER**: Ilham Aliyev **ORG**: _	**LOC**: Azerbaijan, Baku **PER**: Ilham Aliyev **ORG**: _	0.148	0.000	0.500	2.959	2.500
1483806302 1483770632	**LOC**: Atlanta, GA, Washington, D. C., Capitol Hill, BarackO, America, Georgia, New Jersey **PER**: John Lewis, Lewis, RepJohnLewis, Barack Obama, God, Stacey Abrams, Cory Booker, Jim Crow, Mark Hamill **ORG**: Ku Klux Klan	**LOC**: America, Georgia, Mississippi Delta, Edmund Pettus Bridge **PER**: John Lewis, Peniel Joseph, Jim Crow, Barbara Jordan, Peniel Joseph, Lewis, Crow, Donald Trump, **ORG**: Center for the Study of Race and Democracy, LBJ School of Public Affairs, CNN, University of Texas at Austin	0.078	0.071	0.971	2.492	1.000
1546012672 1488866568	**LOC**: Dresden, Chemnitz **PER**: _ **ORG**: Staatsanwaltschaft	**LOC**: Dresden **PER**: Carolyn, Carolyn Anne Cavender **ORG**: Jackson Madison, General Hospital	0.471	0.500	0.998	3.360	4.000

Table 4: Example of performance of the best NER model. (Huggingface NER extractor, Huggingface vectorizer, Gradient Boosting model).

In this section, we provide some further comments on Table 4. In the Table example output of the best-performing NER approach can be found. You can notice several patterns, which could be the reason for the low quality of prediction of NER approaches. First of all, the errors of the NER tagger makes the model performance worse. You can see several wrong detections. For example, «BarackO» definitely should be tagged as person, not location, in the second example.

Also, in our method, in cases when no named entities were found for the PER, ORG or LOC classes, the distance score was set to 0.5, because it is not clear whether the absence of named entities is an indicator of similarity or not. These 0.5 scores confuse the model, increasing its generalization error. The second problem is that when there is no overlap of named entities in one of the classes, it could lead to two bad outcomes. When the other two distance scores correctly reflect the ground true similarity, like in the second example in Table 4, the one with no overlap could be large, which spoils the overall prediction.

The second behavior happens when the extracted entities have no straight overlap but happen to be similar in vector space. For example, two different news about the close locations. In this case, the model can output a small distance, which is not correct. As a result, the model tends to predict values from the middle of the $[1, 4]$ range, avoiding its edges. However, there are examples which show good performance. In the third example, there is an overlap in one word for LOC, which gives the distance from the middle of the range. There is no overlap in organizations. As a result, we get a score quite similar to ground true, taking into account the peculiarities discussed above.

Tagger	Vectorizer	Linear Regression	SVR	Decision Tree	Random Forest	Gradient Boosting
Huggingface	BOW	0.202	0.200	0.154	0.244	0.246
	Tf-Idf	0.195	0.191	0.135	0.229	0.239
	Fasttext	0.194	0.194	0.157	0.320	0.326
	Huggingface	0.250	0.250	0.200	0.385	**0.395**
Polyglot	BOW	0.228	0.227	0.146	0.240	0.244
	Tf-Idf	0.220	0.218	0.143	0.227	0.226
	Fasttext	0.206	0.205	0.151	0.309	0.310
	Huggingface	0.211	0.211	0.180	0.334	**0.342**
Spacy	BOW	0.227	0.227	0.147	0.230	0.235
	Tf-Idf	0.223	0.223	0.154	0.224	0.231
	Fasttext	0.184	0.183	0.146	0.254	0.259
	Huggingface	0.219	0.220	0.152	0.278	**0.279**

Table 5: Comparison of different NER taggers, vectorizers and ML models for evaluation dataset. Pearson correlation.

	Transformer-EncoderCLS	Transformer-EncoderCosSim
distilbert	0.591	0.679
bert-base-cased	0.644	0.704
bert-base-uncased	0.678	0.714
xlm-roberta-base	0.656	0.643
xlm-roberta-large	**0.706**	0.718
xlm-mlm-17-1280	0.650	**0.734**

Table 6: Comparison of performance of different pre-trained encoders from Transformers on evaluation dataset. Pearson correlation.

	Validation	Evaluation
LinearRegression	0.290	0.364
SVR	0.288	0.356
DecisionTreeRegressor	0.228	0.273
RandomForestRegressor	0.477	0.469
GradientBoostingRegressor	**0.478**	**0.477**

Table 7: Comparison of performance of different pre-trained encoders from Transformers for «NLI pairs - titles» approach. Pearson correlation.

	Validation	Evaluation
Polyglot	0.461	0.342
Spacy	0.426	0.279
Huggingface	**0.496**	**0.395**

Table 8: Comparison of the results on cross-validation and evaluation dataset. Pearson correlation.