HFL at SemEval-2022 Task 8: A Linguistics-inspired Regression Model with Data Augmentation for Multilingual News Similarity

Zihang Xu[†], **Ziqing Yang**[†], **Yiming Cui**[‡†], **Zhigang Chen**[†]
[†]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China
[‡]Research Center for SCIR, Harbin Institute of Technology, Harbin, China
[†]{zhxu13,zqyang5,ymcui,zgchen}@iflytek.com
[‡]ymcui@ir.hit.edu.cn

Abstract

This paper describes our system designed for SemEval-2022 Task 8: Multilingual News Article Similarity. We proposed a linguistics-inspired model trained with a few task-specific strategies. The main techniques of our system are: 1) data augmentation, 2) multi-label loss, 3) adapted R-Drop, 4) samples reconstruction with the head-tail combination. We also present a brief analysis of some negative methods like two-tower architecture. Our system ranked 1st on the leaderboard while achieving a Pearson's Correlation Coefficient of 0.818 on the official evaluation set.

1 Introduction

In Task 8 (Chen et al., 2022), we are expected to assess the similarity of pairs of multilingual news articles as shown in Table 1. Ten different languages are covered in this task, including Spanish, Italian, German, English, Chinese, Arabic, Polish, French, Turkish and Russian. Task 8 emphasizes more the events themselves described in the news rather than the style of writing or other subjective characteristics. Therefore, it is beneficial to improve the quality of clustering of news articles and to explore similar news coverage across different outlets or regions.

The foundation model (Bommasani et al., 2021) we choose is XLM-RoBERTa (XLM-R) (Conneau et al., 2019) which has been proved to be a powerful multilingual pre-trained language model compared with other models like mBERT (Devlin et al., 2018) and it can process all the languages existing in Task 8. Based on that, a great variety of strategies have been tested along with our exploration like data augmentation (DA), head-tail combination, multi-label loss, adapted R-Drop, etc.

Through this task, we realized the importance of data quality and efficient training schemes in such a cross-lingual setting. By struggling to improve the richness of the data and find out what methods are effective when training such a similarity assessment model, our system[1] ranked 1st in this competition.

2 Background

2.1 Dataset Description

There are 4,964 samples with 8 language pairs in the training set and the test set contains 4,593 samples in 18 different language pairs, the details of which are presented in Table 2. Due to some inaccessible URLs, the training set is slightly smaller than it should be (22 samples missing in total).

The similarity scores of pairs of articles in the provided dataset are rated on a 4-point scale (between 1 and 4) from most to least similar in 7 subdimensions, including *Geography*, *Entities*, *Time*, *Narrative*, *Overall*, *Style* and *Tone* (an example is provided in Appendix). However, only the predictions for *Overall* will be used to evaluate the performance of our systems.

2.2 Related Work

Research on text similarity always attracts people's eyes as it acts as the basis of quite a few NLP downstream tasks like information retrieval (Ponte and Croft, 2017). Previously, some methods based on statistics like BM25 (Trotman et al., 2014) and Edit Distance (Ristad and Yianilos, 1998) are used to evaluate the relevance between two texts but they do not work anymore in cross-lingual settings. Then, after dense word embedding in low dimensions like Word2Vec (Mikolov et al., 2013) was put forward, methods like calculating the cosine similarity (Rahutomo et al., 2012) with the sentence embedding based on each word embedding came into use. However, it is hard for these approaches to capture the latent meaning of the whole article precisely. Nowadays, depending on transformer-based

[1]Our codes are available at https://github.com/GeekDream-x/SemEval2022-Task8-TonyX

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1114 - 1120

Key	Value
Pair_id	1626170156_1623571850
Lang1/Lang2	de/en
News1	US-Bürgerrechtler verklagen Trump wegen Polizeieinsatzes. Der Einsatz am Montag sei gesetzwidrig gewesen, da die Demonstranten sich friedlich verhalten hätten, ... Tod des Afroamerikaners George Floyd bei einem brutalen Polizeieinsatz in Minneapolis ausgelöst worden. Im Zuge der Proteste kam es immer wieder zu Ausschreitungen.
News2	Joe Biden Addresses The Nation On Race And Trump's Attacks On Protesters Via the Washington Post: Seeking to console a nation riven by nights of violence with a promise to heal its racial wounds, ... — "I can't breathe" — as a mantra. Floyd, an unarmed black man, died after a police officer knelt on his neck in Minneapolis.
Scores	**Geography** 1.0 **Entities** 2.0 **Time** 1.0 **Narrative** 2.0 **Overall** 4.0 **Style** 2.0 **Tone** 1.0

Table 1: An example in the training set.

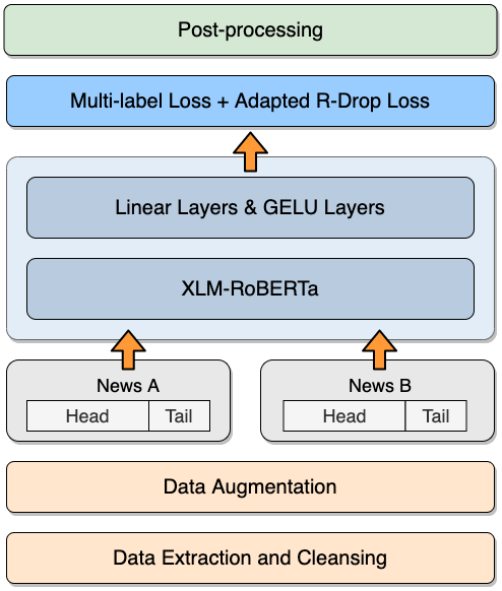

Figure 1: The overall framework of our system proposed for SemEval-2022 Task 8.

general pre-trained models are becoming the new paradigm and plenty of models for multilingual and cross-lingual settings have been proposed like mBERT (Devlin et al., 2018), ERNIE-M (Ouyang et al., 2020) and XLM-R (Conneau et al., 2019).

3 System Overview

Our baseline system is simply providing a pair of articles to XLM-R and regressing its output from [CLS] token to the manually annotated similarity score by training with Mean Squared Error (MSE). All the optimization methods discussed below are applied based on this architecture and the overall framework of our final system is illustrated in Figure 1. After training with all the positive strategies, we then made an ensemble of the best models on each fold for the final prediction.

3.1 Data Augmentation

In this task, we augmented the training data in two different ways and they will be introduced respectively in the following subsections.

3.1.1 Back Translation

It is clear from Table 2 that the original training set is not sufficient to train XLM-R, so we made use of back-translation to enrich it. As the English pairs account for the largest, we only paid attention to the non-English samples in this stage. Take the French samples for example, by calling Google Translation API[2], we translate the French articles to English and then translate the English texts back to French. As for the cross-lingual pairs with German and English, we only back-translate the German part and then combine it with the corresponding English part to form a new sample.

3.1.2 Translate Train

Another weakness of the original training set is the severe lack of some monolingual language pairs which exist in the test set but not in the training set like Chinese and quite a few cross-lingual language pairs like German to French. To deal with this problem, we planned to generate translate-train data to fill the gap.

In such semantic comprehension tasks, it is undoubted that the richer semantic information is, the better the model performance will be. Therefore, for maintaining the semantic richness to the largest extent, we made an arrangement for the construction of the translate-train set (details are provided in Table 3).

As the average quantity of non-English monolingual samples in the training set is 430, for the sake

[2] https://cloud.google.com/translate

	ar	de	en	es	fr	it	pl	ru	tr	zh	de-en	de-fr	de-pl	es-en	es-it	fr-pl	pl-en	zh-en	Total
Train	274	857	1787	567	72	0	349	0	462	0	574	0	0	0	0	0	0	0	4942
Test	298	611	236	243	111	442	224	287	275	769	190	116	35	498	320	11	64	223	4953
Train+DA	548	1714	1787	1134	461	586	689	401	924	800	1148	317	0	586	586	0	349	800	12830

Table 2: Data distribution in each set. Columns with one language (e.g. "zh") mean the two articles in a pair are in the same language. Columns with two languages (e.g. "zh-en") indicate the corresponding cross-lingual pairs.

Origin	Quantity	Target
en-en	401	ru-ru
	800	zh-zh / zh-en
	586	it-it / es-en / es-it
pl-pl	349	pl-en
de-en	317	de-fr / fr-fr

Table 3: Arrangement for the construction of translate-train set.

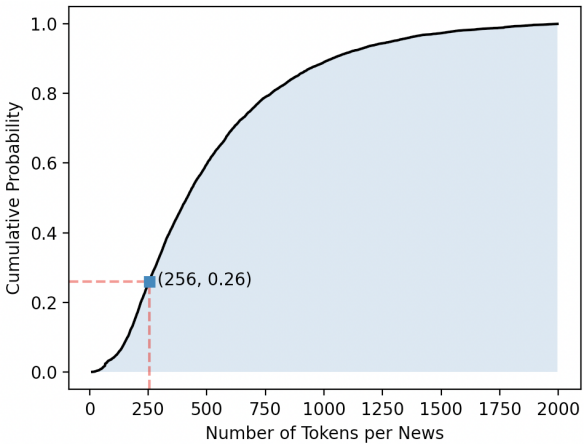

Figure 2: Cumulative probability distribution of article lengths in the training set.

of balancing the whole dataset, we decided to round it down to 400 and let it be the number of translated samples for Russian (due to some precision issues, it became 401 accidentally). As we may know, Russian and English both belong to Indo-European Family (Fortson IV, 2011) while Chinese is a member of the Sino-Tibetan Family (Thurgood and LaPolla, 2016), which indicates that there are quite a lot of common characteristics between the two languages like syntactic structures and lexical analysis methods. So, the most English samples in the original training set would help more in understanding Russian instead of Chinese. Therefore, we decided to generate more Chinese pairs and here we just doubled the number for Russian. Furthermore, the English samples left were all used for generating samples in Italian and Spanish.

In order to improve the reusability of those samples newly translated already, some work on recombination among different languages pairs was done in this phase. For instance, translating German to English samples to French would let us get German to French samples in the meantime.

3.2 Head-tail Combination

There is no doubt that different types of texts have different features. As for news, the title tends to be the most informative place in each article since the authors need to use as few concise words as possible to let the readers know what happened in the story. Besides, we believe the head and tail parts of a news article provide much information as well as similar to the introduction and conclusion parts in a research paper. As the XLM-R is capable of processing 512 tokens in each sequence (a pair of articles) at most and the large majority of articles in the training set are much longer than 256 tokens (see Figure 2), we tried different truncation strategies to further boost the model performance.

3.3 Multi-label Loss

As introduced in Background, only the predictions for *Overall* will be used to evaluate, but the other 6 sub-dimensions are also probably helpful for assisting in building a better model. Consequently, we tried to assign various weights for *Overall* when calculating the loss while treating other sub-dimensions equally. For example, if the loss for *Overall* accounts for 40%, the percentages of the other six sub-dimensions are all 10% individually.

3.4 Adapted R-Drop

R-Drop is proved to be an effective regularization method based on dropout, by minimizing the KL-divergence of the output distributions of every two sub-models generated via dropout in model training (Liang et al., 2021). To better fit with this regression task, we replaced the KL-divergence loss with MSE loss (adapted R-Drop). Similarly, at each training step, we feed the samples through the forward pass of the network twice. Then, our adapted R-Drop method tries to regularize the model by

minimizing the two predicted scores for the same sample, which is:

$$L_R^i = \text{MSE}(y_1^i, y_2^i)$$

where the y_1^i means the model output in the first forward pass for the i_{th} sample. With the basic MSE loss L_B of the two forward passes:

$$L_B^i = \frac{1}{2} \cdot (\text{MSE}(y_1^i, \hat{y}^i) + \text{MSE}(y_2^i, \hat{y}^i))$$

where the $\hat{y}^i$ is the label of the i_{th} sample, the final training target is minimizing L^i for i_{th} sample:

$$L^i = \alpha \cdot L_R^i + (1 - \alpha) \cdot L_B^i$$

where the α is the weight for the adapted R-Drop loss. Based on this introduction, it is easy to extend the formulas to those of forwarding three times.

3.5 Extra Linear Layers

In our baseline system, the prediction score is generated by passing the output of [CLS] token from XLM-R through a single linear layer with the size of (1024, 1). In other words, there are only 1024 parameters that are responsible for the regression from the sentence representation vector to the prediction score, which is probably beyond their power. Hence, we attempted to add a few more layers on top of the XLM-R.

3.6 Post-processing

Once getting the prediction scores, we further corrected some wrong numbers which were outside the expected range. As introduced in Section 2.1, the annotators annotated the similarity in the range (1, 4); consequently, we clipped the outliers.

4 Experimental setup

4.1 Dataset Split

Both the original training set and the training set with DA set were split into 10 subsets with no intersection by random sampling. All the experiments discussed in this paper were conducted with 10-fold cross-validation, and the results displayed are the averages. By using the cross-validation method (Browne, 2000), we could ensure the strategies applied will take a good effect on the final test set to the largest extent.

System	Pearson's CC
w/ data augmentation	
Baseline	83.49
+ DA	**85.86**
w/o data augmentation	
Baseline	84.94
+ Head-tail Combination	85.38
+ Multi-label Loss	85.33
+ Adapted R-Drop	**86.14**
+ Extra Linear Layers	85.50

Table 4: Best results with training methods we used.

4.2 Pre-processing

The news articles in all the data sets are released as URLs and the task organizers offer us a python script[3] which helps to download the pages. After downloading the original files in JSON format, we then extracted and combined "title" and "text" parts of each article and abandoned all other information like "description". Before starting training our model, apart from conducting data augmentation to the training set, we also cleaned the data and joined the head and tail parts of each article. During the process of cleaning, we mainly removed some dirty formatted data like URLs and file paths.

4.3 Evaluation Metrics

The evaluation metric for task 8 is the Pearson's Correlation Coefficient (Pearson's CC) which is a measure of linear correlation between two series of data with a range from -1 to 1 (from least to most correlated) (Stigler, 1989).

4.4 Others

Although hyper-parameters tuning is not a crucial point in our work, we tested a few values for several of them within a small range as they did have an influence on our decisions about how well a strategy worked (see Appendix). Additionally, to help readers replicate our experiments, the details of tools and libraries are provided (see Appendix).

5 Results

5.1 Overall Performance

Finally, our system got 0.818 on the evaluation set according to the official scoring system and

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

Head	Tail	Pearson's CC
256	0	84.94
200	56	**85.38**
128	128	85.21
56	200	84.53
0	256	78.85

Table 5: Results on different head-tail combinations.

ranked 1st. As results are shown in Table 4, all the strategies introduced in Section 3 turned out to have positive effects, and we will discuss the effect of the strategies mentioned individually in the following subsections. For convenience, all the results from our experiments are multiplied by 100.

5.2 Data Augmentation

To find out whether the augmented data was helpful or not, we trained our system on the original training set and the training set with DA respectively (samples used for testing were removed in both of them), then tested it on each fold of the DA set. In experiments on other strategies, we trained and tested our system on the original training set. And this is the difference between the two baselines in Table 4.

Without any surprise, an evident increase is observed from the results displayed in the top part of Table 4, based on which we could make a conclusion that a more abundant training set is definitely beneficial for building a strong system.

5.3 Head-tail Combination

As introduced in Section 3.2, we realized the importance of the head and tail parts of the news articles. However, we cannot determine which part should be paid more attention to heuristically. So, we tried on different ratios of head-tail combination and the results are enumerated in Table 5. Clearly, the head part plays a much more important role by comparing the first and last rows where only either of them are used. However, from the middle three rows where the head and tail parts are combined, it is indicated that the tail part also benefits the whole model performance.

5.4 Multi-label Loss

As discussed in Section 3.3, we used other 6 dimensions and assigned a few different values for the weight of *Overall* from 0% to 100%. It is explicitly

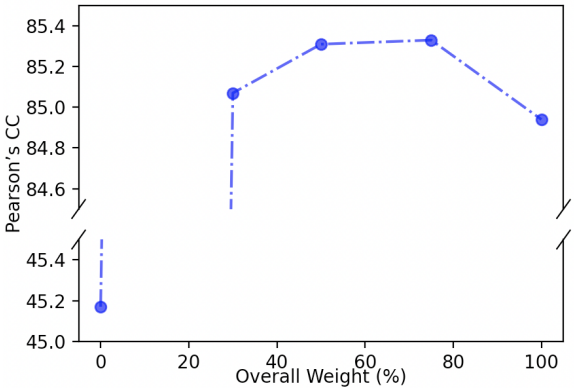

Figure 3: Results on training with multi-label loss.

observed from Figure 3 that there is an overwhelming increase followed by a slight drop while the weight of *Overall* rises gradually. Based on the experiment results, we believe that *Overall* is of the greatest importance to this task, yet the other 6 subdimensions also have a positive effect on achieving a better similarity assessment system.

5.5 Adapted R-drop

As described in Section 3.4, the training loss in our system is composed of both the loss between predictions and labels and the loss between the predictions from different forwarding processes. Here, we explored forwarding once to three times while changing the weight of adapted R-Drop loss.

Apparently, there is a phenomenon from Figure 4 that no matter how large the weight of R-Drop loss is, the more forwarding times are, the better results we will achieve. However, by comparing the results between forwarding once and twice and the results between forwarding twice and three times, we speculate that there is a marginal utility (Kauder, 2015) on this trick, which means the additional benefit from this method will decrease while simply continuing increasing the number of forwarding.

5.6 Extra Linear Layers

During the process of exploration in this direction, we attempted to add 2 or 3 extra linear layers to test if it worked. In the 2-layer setting, the sizes of the layers are (1024, 512) and (512, 1) while sizes composed of (1024, 768), (768, 256) and (256, 1) are prepared for the 3-layer setting. Two sets of experiments were conducted in both settings about whether to put an activation layer (we used GELU (Hendrycks and Gimpel, 2016) here) between adjacent linear layers or not.

It can be observed from Table 6 that there is only

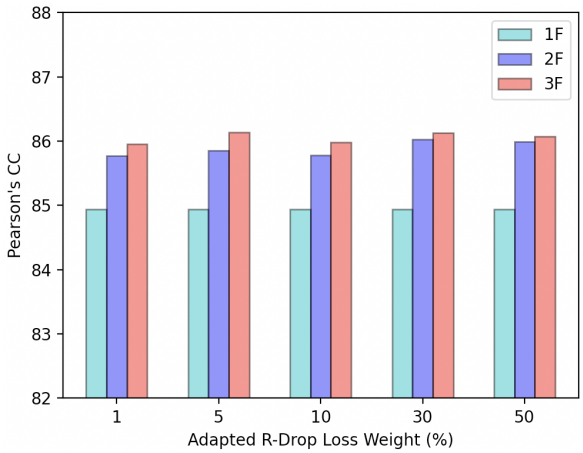

Figure 4: Results about adapted R-Drop (RD) in different settings. "2F" means forwarding twice.

System	Pearson's CC
1-layer	84.94
2-layer	85.46
+ activation	**85.50**
3-layer	85.32
+ activation	85.23

Table 6: Results on different extra layers.

a quite small difference that caused by activation layers in each setting and the effect of that is not always positive. In addition, by comparing the results from different settings, we could draw a conclusion that more parameters did help to boost the system performance even if the benefit does not show linear growth.

5.7 Negative Results

Aside from the strategies discussed above, several tricks that were attempted to deploy in our system as well turned out to be meaningless or had a bad effect on the model performance. For example, we tried to use a pooling vector (max or mean) or the fusion of [CLS] vectors from different layers in XLM-R as the article representation. We also tried to expand the length of sentences that XLM-R could process to 1024 tokens by modifying its position embedding matrix by means of adding a random shift vector after each vector or just randomly initializing the latter part of the learnable expanded matrix. Each negative strategy mentioned above brought approximately at least 2 points drop on the Pearson's CC. Furthermore, unsurprisingly, a two-tower architecture where each shared-parameter model processed each article in a pair led to scores of points decrease, which reflected the importance of semantic interaction via the attention mechanism inside the model.

en	de	es	pl	tr
87.19	84.96	86.64	75.29	83.54
ar	**ru**	**zh**	**fr**	**it**
79.42	78.47	76.78	86.53	86.17
es-en	**de-en**	**pl-en**	**zh-en**	
86.35	85.98	88.18	81.00	
es-it	**de-fr**	**de-pl**	**fr-pl**	
81.97	68.89	64.31	82.68	

Table 7: Individual results of all language pairs in our best submission.

5.8 Error Analysis

After the evaluation phase ended, the evaluation data with labels were provided and we also checked the system performance on different language pairs individually. The details of our best submission are presented in Table 7. It is obvious that the model tends to perform worse on the language pairs which are rare or absent from the training set like German to Polish (only 64.31). Interestingly, although having seen monolingual samples in Polish and related cross-lingual data, the system still behaves badly on Polish monolingual data (just slightly over 75), which is probably due to its complicated lexical variation and grammar rules (Smoczynska, 2017).

6 Conclusion

By deploying various optimization methods, including data augmentation, head-tail combination, multi-label loss, adapted R-Drop and adding extra linear layers, we built a relatively strong system for assessing the similarity between a pair of news articles in multilingual and cross-lingual settings and ranked 1st in the competition with a Pearson's CC of 0.818 on the official evaluation set.

In the future, apart from enriching the training data, we are also supposed to analyze the languages individually and try to leverage the exclusive rules or features of each language rather than relying too heavily on general pre-trained models to further boost the model performance, especially on those minority languages.

References

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Michael W Browne. 2000. Cross-validation methods. *Journal of mathematical psychology*, 44(1):108–132.

Xi Chen, Ali Zeynali, Chico Q. Camargo, Fabian Flock, 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, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

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

Benjamin W Fortson IV. 2011. *Indo-European language and culture: An introduction*. John Wiley & Sons.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Emil Kauder. 2015. *History of marginal utility theory*. Princeton University Press.

Xiaobo* Liang, Lijun* Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks. In *NeurIPS*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-m: enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674*.

Jay M Ponte and W Bruce Croft. 2017. A language modeling approach to information retrieval. In *ACM SIGIR Forum*, volume 51, pages 202–208. ACM New York, NY, USA.

Faisal Rahutomo, Teruaki Kitasuka, and Masayoshi Aritsugi. 2012. Semantic cosine similarity. In *The 7th International Student Conference on Advanced Science and Technology ICAST*, volume 4, page 1.

Hyperparameter	Range/Value
Epoch	20 ~ 30
Batch Size	32
Weight Decay	1e-4
Warm-up Rate	0.1
Learning Rate	5e-6 ~ 3e-5
Overall Weight	0 ~ 1
Adapted R-Drop Weight	0.01 ~ 0.5

Table 8: Main hyper-parameters tuned in our system.

Tools & Libraries	Version
NumPy	1.21.2
pandas	1.2.4
Python	3.7.10
PyTorch	1.9.0
Transformers	4.5.1
semeval_8_2022_ia_downloader	0.1.7

Table 9: Main tools and libraries used in our system.

Eric Sven Ristad and Peter N Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Magdalena Smoczynska. 2017. The acquisition of polish. In *The crosslinguistic study of language acquisition*, pages 595–686. Psychology Press.

Stephen M Stigler. 1989. Francis galton's account of the invention of correlation. *Statistical Science*, pages 73–79.

Graham Thurgood and Randy J LaPolla. 2016. *The sino-tibetan languages*. Taylor & Francis.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 2014 Australasian Document Computing Symposium*, pages 58–65.

A Appendix

Table 8 and Table 9 provide the details about the corresponding hyper-parameters and libraries respectively, which are beneficial to help replicate our experiments.