INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.153

Evaluate Similarity of Requirements with Multilingual Natural Language Processing

U. Bisang [1], J. Brünnhäußer [1,✉], P. Lünnemann [1], L. Kirsch [2] and K. Lindow [1]

[1] Fraunhofer IPK, Germany, [2] CONTACT Software GmbH, Germany

✉ joerg.bruennhaeusser@ipk.fraunhofer.de

Abstract

Finding redundant requirements or semantically similar ones in previous projects is a very time-consuming task in engineering design, especially with multilingual data. Due to modern NLP it is possible to automate such tasks. In this paper we compared different multilingual embeddings models to see which of them is the most suitable to find similar requirements in English and German. The comparison was done for both in-domain data (requirements pairs) and out-of-domain data (general sentence pairs). The most suitable model were sentence embeddings learnt with knowledge distillation.

Keywords: *artificial intelligence (AI), requirements management, information management, data-driven design, natural language processing*

1. Introduction

The goal of automating engineering design tasks or supporting them with algorithms has been pursued since the introduction of virtual product creation. Over the last decade virtual product creation has been evolving and the increasing availability of learning algorithms represents another new potential. The practical application faces various challenges. Among them are data availability, training of models, identification of potential application areas, and support of creative value-added work. In the breadth of the necessary design activities, a distinct potential for the use of learning algorithms can be seen in particular in the area of information provision, processing and comparison.

One of the most important phases of designing a new product is to gather the requirements which are derived from company expectations, previous projects or customer needs. Natural language used in requirements specification can be ambiguous leading to mistakes which are very crucial and cost intensive in the later stages of product development. Additionally, many tasks, like specification clean-up, evaluation and quotation, in early stages of requirements engineering take a long time because it is still mostly a manual task. In many industries, especially in the automotive industry, working with hundreds to thousands of requirements in a project is common. The manual effort of identifying knowledge for quotation or former defined risks of customer requirements in the company is significant, especially in the request for quotation (RFQ) phase. In some cases, requirements are formulated redundantly or even contradictorily in specifications. If not detected, this can lead to increased efforts in processing or to critical aberrations in product design and verification. Another challenge in this area is that multinational companies in particular have to cope with multilingual specifications. Texts may be available in several languages at the same time and have to be kept consistent, or individual specifications may be monolingual and still have to be considered in the multilingual process. One solution presented here is to automate these multilingual time-consuming manual processes, thus reducing costs and increasing process quality at the same time. With the

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

potential of artificial intelligence and natural language processing (NLP), the technological fundamentals exist and need to be applied to the field of specific domain knowledge.

In this context, the paper addresses the challenge of multilingual requirements. It is illustrated how open source NLP algorithms are adapted to provide specific solutions in requirements management. Natural language processing is increasingly affected by the popularity of artificial intelligence and neural networks which are trained on large sets of text data. Particularly modern contextual NLP models are performing very well in practice. In order to save resources some tasks can be assisted or even fully automated by NLP models or word embeddings. In our previous work (Brünnhäußer et al., 2021) the most suitable word embeddings have been examined for the search of semantically similar requirements. In this paper we compared different multilingual NLP models regarding their performance in the task of finding similar sentences or specifically similar requirements. Therefore, the multilingual models are compared to the translation via an online translation service and English-only word embeddings. The central research question of this contribution is to evaluate whether multilingual models perform better than online translation services combined with monolingual embeddings in finding semantically similar requirements and if so, which of the presented models is most suitable. In the following, we investigate this question in principle and not with respect to a specific use case.

If multilingual models work better than translation services, it will enable the knowledge representation via those word embeddings across language boundaries and eventually greatly reduce the effort to work within different languages. In consideration of industrial application, multilingual similarity analysis can support collaborative work. This could help in many design tasks in requirements engineering like finding duplicates in a multilingual dataset or identifying similar requirements from historic projects in order to reuse existing designs.

2. Theoretical background and previous work

Modern natural language processing made a huge advance with the proliferation of neural networks. Word2Vec was one of the first word embedding models which used neural networks and large sets of text data came from Google and (Mikolov *et al.*, 2013). The performance of those word embeddings which converted words to 300 dimensional vectors were much better than previous approaches of NLP and they could be used in many fields like spam recognition or text similarity tasks. A further enhanced model is fastText (Bojanowski *et al.*, 2016) but all of those models are still transferring the words like a dictionary into a specific vector without looking at the words context within the sentence it occurs.

Contemporary word embeddings are mostly contextual which means that the same word is turned into a different vector based on the previous and subsequent words of the sentence. One famous example of contextual word embeddings is BERT (Devlin *et al.*, 2019) and those embeddings improved the capabilities of NLP even further.

Based on those advances several multilingual word embeddings have been developed recently. (Conneau *et al.*, 2020) introduced the Crosslingual Language Model-RoBERTa (XLM-R) which is one of the best multilingual word embeddings according to different benchmarks (Conneau et al., 2020). It is a transformer-based masked language model. Furthermore, it was trained on a corpus of more than two terabyte CommonCrawl data and 100 languages (Conneau *et al.*, 2020). Transformer models (Vaswani *et al.*, 2017) transform input sequences to output sequences but unlike recurrent neural networks not in sequential order. The transformer architecture is a neural network encoder-decoder architecture and its ability to process sequential data is based on the concept of attention. This difference allows a transformer to make connections in multiple directions, while recurrent networks are only able to make connections to previously seen tokens (Vaswani *et al.*, 2017). Hence, the context and the position of the word within the sentence is considered which isn't the case in recurrent neural networks.

Besides word embeddings there are sentence embeddings which transfer a whole sentence into a vector. One model was trained by (Reimers and Gurevych, 2020) with a concept called knowledge distillation introduced by (Geoffrey Hinton *et al.*, 2015). It is a model compression technique which enables the transfer of several or very large models into a single smaller model. The approach of (Reimers and Gurevych, 2020) extends existing sentence embedding models from monolingual to multilingual. It needs a teacher model, a student model and sentences with a parallel translation.

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

During training from English to another language, the student model tries to imitate the teacher model's output embedding for both the English input sentence and the parallel German sentence. Two models were trained which are called in this paper KD-Par for paraphrase and KD-Sim for similarity. KD-Par and KD-Sim were trained in the same way, but with different data and different teacher models: KD-Sim was trained by an SBERT model (Reimers and Gurevych, 2019) trained on data annotated for similarity and is optimised to embed similar sentences into similar dimensions. KD-Par on the other hand was trained by a RoBERTa-based model (Yinhan Liu *et al.*, 2019) and is optimised to embed sentences that are paraphrases of each other into similar dimensions. KD-Par has also seen a larger variety of data.

There are several ways for estimating the similarity of word or sentence vectors. The simplest approach is to calculate the cosine of two vectors which can be between 1 when the words or sentences are semantically similar and 0 which means they aren't similar at all (Goldberg, 2017). Besides the mean another way to calculate a sentence vector from multiple word vectors is the smooth inverse frequency (SIF) (Arora *et al.*, 2019). The SIF is a weighted average, which weights the word embeddings in a sentence according to their overall occurrence, the rarer a word is, the higher its weight becomes. After averaging, the common component is removed. A more advanced approach for comparing the similarity is the Word Mover's Distance from (Kusner *et al.*, 2015) which tries to find the shortest distance between each word of two sentences. The result is a sum of distances between two comparing sentences where 0 means there is no distance at all and the sentences are exactly the same. Another important aspect is the metric for annotating the actual semantic similarity of two statements. (Agirre *et al.*, 2012) created a metric which is used for different benchmarks in order to compare different word embeddings. The STS metric labels are on a scale from 0 to 5, where 0 means the compared sentences are completely different and do not even discuss the same topic and 5 indicates completely equivalent sentences.

One research about the performance of multilingual embeddings was provided by (Sourav Dutta, 2021) and is about cross lingual embeddings in seven different domain-specific applications like medical, religious or legislative documents but not on requirements data. (Sourav Dutta, 2021) uses fastText and the WMD as well and additionally two different BERT models, one of which is SBERT, which was also used as teacher model to train KD-Sim. In this study the WMD with fastText on aligned language embeddings performed best and the SBERT models second-best.

3. Method and experiment design

We are using two different sets of data. The in-domain data consists of requirements but if we want to know how the NLP models perform on requirements as compared to more general sentence pairs we need another general dataset for comparison, the out-of-domain data. In this section we describe the in-domain and out-of-domain data, their preparation and how we annotated it. Furthermore, we present the used models, the way we translated the data for the monolingual models and how we calculated the similarity upon the results. The high-level approach is shown in Figure 1.

Two datasets were used for evaluation, both were generated by the authors. One dataset, the out-of-domain data contains general sentence pairs and was used to proof that the evaluated embeddings are generally suitable for the task of assessing the similarity of sentence pairs. The in-domain data consists of design, software and hardware requirements and was used to test the suitability for comparing requirements pairs. Example sentences for both datasets can be found in Table 1. The datasets had to be generated by the authors, because no human-translated datasets of English and German sentence pairs that are annotated for semantic textual similarity exists and for requirements no public datasets that are annotated for semantic textual similarity seem to exist at all.

The out-of-domain data consists of 300 pairs of general sentences. These sentence pairs were randomly selected from the Cross-lingual Natural Language Inference (XNLI) corpus (Conneau et al., 2018), a corpus consisting of English sentence pairs that are annotated for entailment (the two sentences are contradictory, follow from each other or are neutral to each other) and were human-translated by the original authors to 14 languages, among them German. The 300 randomly selected pairs were annotated by one annotator with the semantic textual similarity score described in the last chapter.

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

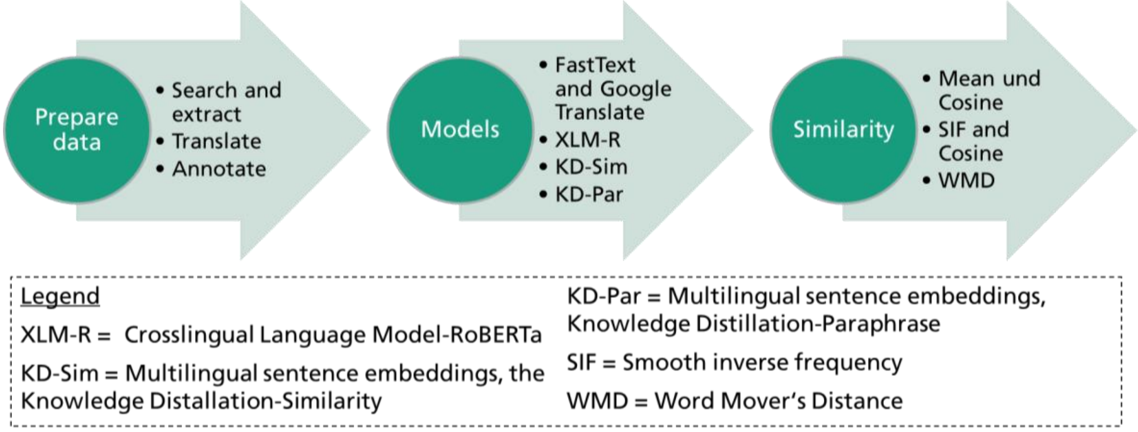
Prepare data
• Search and extract
• Translate
• Annotate
Models
• FastText and Google Translate
• XLM-R
• KD-Sim
• KD-Par
Similarity
• Mean und Cosine
• SIF and Cosine
• WMD
Legend
XLM-R = Crosslingual Language Model-RoBERTa
KD-Sim = Multilingual sentence embeddings, the Knowledge Distallation-Similarity
KD-Par = Multilingual sentence embeddings, Knowledge Distillation-Paraphrase
SIF = Smooth inverse frequency
WMD = Word Mover's Distance

Figure 1. The steps of the approach with their details.

The in-domain dataset is based on the PURE corpus of real requirements documents (A. Ferrari *et al.*, 2017). The requirements were extracted from the documents using image processing tools. First images of each page were binarized into black and white images, next dilation was used to determine sections, the content of which was then extracted using optical character recognition. Then the sections were classified based on their position and content into containing a requirement text, containing additional information to requirements or being unrelated to requirements. The sections that were identified as containing a requirement text were used to build the in-domain corpus.

These extracted requirements were first combined into pairs using similarity bands. 100 pairs in each of the following similarity bands: 0.82 - 0.85, 0.85 - 0.88, 0.88 - 0.91, 0.91 - 0.94 and 0.94 - 0.97 were randomly selected. The similarity score for sorting the embeddings into the bands was computed by comparing the fastText embeddings (Grave *et al.*, 2018) of the sentences, averaged with the arithmetic mean, with the cosine similarity as a measure. This approach was inspired by (Agirre *et al.*, 2016), although they additionally use the surface lexical similarity for pairings, this measure takes into account the common words between the sentences.

These 500 sentence pairs were annotated by the same annotator. Still only 32 of the 500 pairs had a non-zero annotation, these 32 pairs and 50 randomly chosen zero-annotated ones were selected for the in-domain data. To have more requirements pairs and more pairs with a higher STS score, 83 new, more similar pairs were selected manually, such that the in-domain data now consists of 165 requirements pairs.

In a final step the in-domain data was translated to German, using a machine translation service to generate an initial translation, which was corrected by hand. This path was chosen as a trade-off between the efficiency of machine translation and the quality of human translation.

To give an example of the two datasets Table 1 contains an example-pair of each of the datasets:

Table 1. Examples for out-of-domain and in-domain data

Data	Sentence 1	Sentence 2	Similarity
Out-of-domain	The one thing I am most proud of is that the IRT is a leader across the country in providing theatre experiences for students.	The IRT is involved in the theater for middle schoolers.	2
In-domain	The system must allow users to easily navigate through each of the help topics by selecting corresponding hypertext link provided in the left frame of the pop-up Help window.	Design Constraints online User Documentation and Help System Requirements relevant, online documentation for users should be available on each page. users must have easy access to help while interacting with the system. Adequate user documentation should be provided to minimize the number of calls to the Help Desk about problems with the system. Modifications should be reported via the main page to inform actors unexpected changes. This electronic documentation should be supplemented with phone and on-site support provided by the Office of Information Services.	1

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

Three different types of embeddings were tested: KD-Par and KD-Sim (Reimers and Gurevych, 2020), two multilingual sentence embeddings models, XLM-R (Conneau *et al.*, 2020), a multilingual word embeddings model and fastText (Grave *et al.*, 2018), a monolingual word embeddings model. To simulate multilingual capabilities for the English-only fastText embeddings, Google Translate[1] was used to translate non-English sentences. All embedding models were used out of the box. The embedded data was compared with two different comparison methods: The cosine similarity and the Word Mover's Distance, turned into a similarity. Since the cosine similarity is only able to compare two vectors, the word embeddings had to be transformed into sentence embeddings, a common approach for this is to use the arithmetic mean. We used both the arithmetic mean and the smooth inverse frequency. For the Word Mover's Distance it is not necessary to aggregate the word embeddings, as it needs the original word embeddings as inputs. Therefore, it is not suitable to generate similarities for the sentences embedded with the sentence embeddings.

In a final step the similarities between the (aggregated) embeddings were compared with the human-generated similarity scores on the datasets, using Spearman's rank correlation. Spearman's rank correlation is used to calculate the correlation or relationship between the ordering of the elements of datasets. The higher this correlation is, the better is the embedding-similarity combination able to emulate human perceptions of similarity between requirements. Spearman's rank correlation is related to the Pearson correlation, which is favoured by the STS tasks (Agirre *et al.*, 2012), but unlike the Pearson correlation, Spearman's rank correlation does not expect the input data to be normally distributed and is less sensitive to outliers, therefore it seems a more suitable measure. Spearman's rank correlation is also chosen by (Reimers and Gurevych, 2020) to evaluate their semantic textual similarity results. In addition to the Spearman's rank correlation, the p-values resulting from a Student's t test on the results of the correlations will also be considered. Here the p-value indicates the likelihood that some correlation would have been found, if the two distributions were not actually related. For example, a p-value of 0.01 means that there is a likelihood of 1% that this coefficient was reached, although the data is not actually correlated. Values with a p-value above 0.01 will be marked in the results and be considered not trustworthy.

4. Results

In this section the results of the approach described above will be outlined. Table 2 and Figure 2 show the average Spearman's rank correlation between the in-domain and out-of-domain data.

Table 2. Average Spearman's rank correlation for out-of domain and in-domain results for different language combinations; green = best result of row (en = English, de = German)

	KD-Par	KD-Sim	XLM-R		fastText		
			Mean	SIF	Mean	SIF	WMD
en-en	57.3	56.1	29.0	42.7	22.3	39.0	42.9
de-de	54.5	54.6	29.9	41.0	11.7	20.8	35.6
en-de	57.7	55.3	19.5	32.1	9.6	27.8	20.3
de-en	52.6	53.4	22.1	39.1	9.4	31.7	21.4
mean	55.5	54.8	25.1	38.7	13.3	29.8	30.0

The best result per row is coloured green, p-values do not apply for this table, as it is an average. It can be seen that KD-Par generally performs best of the compared methods, while KD-Sim is a close second. The similarities of KD-Par's and KD-Sim's embeddings tend to have a moderately high correlation with the gold label similarities. XLM-R with cosine similarity and the smooth inverse frequency for aggregation performs quite well, too. They score quite badly when the similarities are computed using cosine similarity and the arithmetic mean. The fastText embeddings generally perform worst, although using the SIF for aggregation or the Word Mover's Distance for similarity

[1] https://translate.google.com/

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

generally improves their performance. It is noticeable that the fastText embeddings always perform markedly worse for language combinations which contain German, therefore language combinations that necessitate the use of Google Translate.

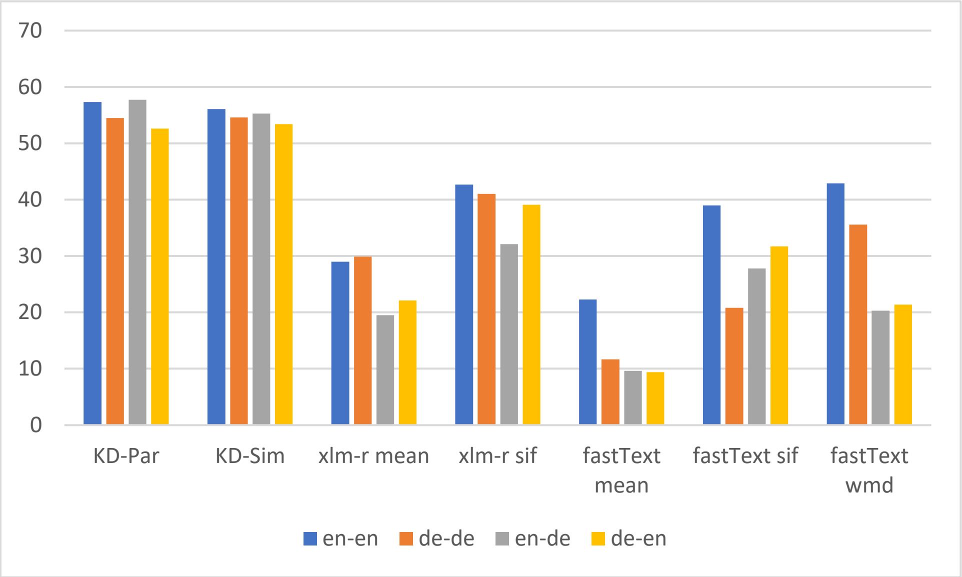

Figure 2. Average Spearman's rank correlation for out-of domain and in-domain results based on Table 2 (en = English, de = German)

Table 3 and Table 4 illustrates the non-averaged results for the in-domain and out-of-domain data, as these values are not averaged, the results with a too high p-value are marked yellow. As for the averaged results, the multilingual sentence embeddings always perform best. KD-Sim performs better for the in-domain data than the out-of-domain data, when it is compared to KD-Par.

Table 3. Average Spearman's rank correlation for out-of-domain data; green = best result of row, yellow = p-value higher than 0.01 (en = English, de = German)

	KD-Par	KD-Sim	XLM-R		fastText		
			Mean	SIF	Mean	SIF	WMD
en-en	57.1	54.9	22.6	37.3	24.1	29.6	40.0
de-de	52.4	52.1	22.2	32.0	14.9	20.9	35.1
en-de	57.0	55.4	18.4	24.8	20.9	23.0	38.7
de-en	51.3	51.5	22.5	27.1	13.5	23.3	35.3
mean	54.4	53.5	21.4	30.3	18.3	24.2	37.3

Table 4. Average Spearman's rank correlation for in-domain data; green = best result of row, yellow = p-value higher than 0.01 (en = English, de = German)

	KD-Par	KD-Sim	XLM-R		fastText		
			Mean	SIF	Mean	SIF	WMD
en-en	57.6	57.3	35.5	48.1	20.5	48.5	45.8
de-de	56.7	57.1	37.6	50.0	8.6	20.7	36.1
en-de	58.3	55.2	20.6	39.3	-1.8	32.5	1.9
de-en	53.9	55.3	21.7	51.1	5.3	40.1	7.5
mean	56.6	56.2	28.8	47.1	8.1	35.4	22.8

An interesting aspect of the performance of the fastText baseline is the difference in performance of fastText between English-English and the other language combinations containing German. This

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

difference is larger for the in-domain data than for the out-of-domain data, even though fastText sometimes performs better on the in-domain data than the out-of domain data.

5. Discussion

From the results described above we can tell that monolingual embeddings in combination with machine translation are not the best performing approach. The most promising approach for this task seems to be the multilingual sentence embeddings. They perform generally well and consistently when comparing different sentences in different language combinations.

XLM-R with smooth inverse frequency and the cosine similarity generally performs acceptably and occasionally well, but generally its performance is rather variable (In the average results, the difference between the best and worst performance is around 11 points, for KD-Par this is only around 5 and for KD-Sim around 3 points). It is possible that the smooth inverse frequency is less suitable for context-aware embeddings like XLM-R than it is for the non-context-aware embeddings it was originally proposed for. XLM-R using the cosine similarity and mean generally performs less well. Further experiments imply that this might be due to the fact that XLM-R with mean and the cosine similarity is better suited to finding direct translations than to finding similar, but not translated sentences, at least without fine-tuning.

The results for fastText imply that Google Translate does not to perform very well for translating the data, as the Spearman correlation for language combinations containing German and therefore translated data are lower than for the English-English setting. An exception to this rule is if fastText is used with the Word Mover's Distance, here all settings with a low enough p-value perform quite similarly.

The worse performance for settings with German data might be due to the fact that the in-domain data is too specific. Other reasons might be that they are not cleaned well enough or the requirements might be too long to translate well. And while the fastText embeddings tend to work well for assessing the similarity for sentence pairs in the out-of-domain data and the English-English settings, they perform worse in settings that include translations and requirements, which makes fastText with the cosine similarity unsuitable for the task at hand. This worse performance might also be due to fastText itself, but since it does consistently well for the English-English setting it seems more probable that the cause is an insufficient machine translation.

FastText with the Word Mover's Distance performs almost as well as XLM-R with the smooth inverse frequency and cosine similarity, it still performs worse than the sentence embeddings. The Word Mover's Distance also has multiple draw-backs: It is not suitable to be used with the sentence embeddings, as it needs embedded words or tokens as input and it is significantly slower than the cosine similarity. Here its similarity performance compared to the multilingual sentence embeddings does not justify using it, when considering its slowness. Still it is noteworthy that assessing the similarity of sentences seems to profit from weighting rarer words more highly than more common ones and therefore using the SIF over the arithmetic mean. It profits from connecting each word to the most similar word in the comparison-sentence or specifically from using the WMD, too.

Even though the sentence embeddings perform best in this paper, they perform worse than they do for the tests in (Reimers and Gurevych, 2020), there KD-Par reaches a Spearman's rank correlation of 83.7 and KD-Sim of 77.9 on the multilingual STS 2017 dataset, while here KD-Par reaches an average Spearman's rank correlation of 57.7 in the best setting and KD-Sim of 55.3.

There are multiple explanations for this difference. (Reimers and Gurevych, 2020) fine-tune their models on the training set of the STS data before testing for it. On the other hand, they also caution that the STS data on which they test is of quite high quality and performance on it might therefore not represent the performance of the sentence embeddings on other, less high-quality data. The in-domain data is generated with optical character recognition and although it was cleaned thoroughly it is still less clean than the STS data. In addition, requirements are often longer and more complex than the sentences in the STS data, therefore it can be considered less easy to process than the STS data. Furthermore, the language of requirements is quite different from every-day language.

Still this data is not entirely representative of requirements data, as the domain from which the requirements data was sourced could have been more expansive. This would have made it possible to test for more of the peculiarities of requirements language. Similarly, if half of the requirements pairs

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

were originally in English and the other half in German, the in-domain dataset would have been able to catch the differences in the style of writing requirements between the two languages. These issues were caused in part by the fact that machine-readable requirements that can be uniquely identified as such are rarely freely available, and written as prose must represent legal and technical facts, so requirements often exist only in product requirements documents.

One more caveat to the results is that the test data is rather small and annotated by one person. A larger test set annotated by more people would have made the results more certain and would have reduced variance and the likelihood that they are a coincident. But creating a larger test set would have been time-intensive and due to budget constraints was out of the project scope.

English and German are two closely related languages, which should be kept in mind when interpreting these results, as more distantly related or unrelated language pairs might lead to quite different results. Still (Reimers und Gurevych 2020) claim that their sentence embeddings have only an insignificant level of language bias. Language bias is a concept introduced by (Roy et al. 2020), which describes the preference of a machine learning model for one language or language combination and therefore performing better when confronted with those. With no language bias a model should not discriminate between language pairs of different languages compared to calculating similarities within one language, which means KD-Sim and KD-Par should perform similarly in and between different languages. The languages that were paired with English for (Reimers und Gurevych 2020)'s test were Spanish, Arabic, Turkish, German, French, Italian and Dutch. All languages in the test except Arabic and Turkish belong to the Indo-European language family and most are still rather closely related to English, but generally this is encouraging for extending our results to more, less closely related languages. Still the performance of additional languages should be tested before deployment and could be worse than the results presented here, whether the additional languages are related to English or not.

6. Conclusion

In conclusion we find that using the cosine similarity calculated from the multilingual sentence embeddings KD-Par or KD-Sim presented by (Reimers and Gurevych, 2020) is the most suitable approach from the ones presented above to find similar requirements in our bilingual data and works better than using monolingual fastText embeddings with Google Translate.

Away from the technological enablement in the use case, this research shows an approach to use artificial intelligence in engineering activities and its potential. It becomes clear that there is a high application potential in particular where information has to be provided and compared. It also shows that the available models can be used in engineering and can also be used without extensive training of the models. Thus, one of the limiting obstacles, the training of models, is eliminated at least in this application.

Specifically, it helps the engineer to focus on more important creative tasks like designing instead of spending time on information management tasks like searching for knowledge for quotation or cleaning requirements. Furthermore, our results enable other automation tasks like searching for relevant information in multilingual datasets which supports the engineer even more in requirements management.

Our research took only English and German into account and we can't make assumptions how well multilingual models work with languages that are unrelated, for example English and Chinese. Furthermore, cultural differences might have to be considered in such datasets as well. These would be interesting topics for future research if suitable research datasets can be found. Another possible next step could be to fine-tune either KD-Par, KD-Sim or both for the domain of requirements data, as fine-tuning seems to have improved the performance for (Reimers and Gurevych, 2020). The fine-tuning should be quite feasible, as the code for the multilingual sentence embeddings is public and very well documented, although the finding of suitable requirements for training might be difficult. After that the fine-tuned model could be included into the CONTACT Elements platform, which would make it possible to validate the approach in practice and test our results in a real design situation.

Acknowledgements

The authors would like to thank Dr. Sharid Loáiciga for her supervision and advice in the framework of a project at the University of Potsdam on which this paper is based.

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

References

A. Ferrari, G. O. Spagnolo and S. Gnesi (2017), “PURE: A Dataset of Public Requirements Documents”, available at: https://www.researchgate.net/publication/320028192_PURE_A_Dataset_of_Public_Requirements_Documents.

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G. and Wiebe, J. (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), Association for Computational Linguistics, San Diego, California, pp. 497–511.

Agirre, E., Cer, D., Diab, M. and Gonzalez-Agirre, A. (2012), “SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity”, in *SEM 2012: The First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Association for Computational Linguistics, Montréal, Canada, pp. 385–393.

Arora, S., Liang, Y. and Ma, T. (Eds.) (2019), A simple but tough-to-beat baseline for sentence embeddings.

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2016), Enriching Word Vectors with Subword Information, available at: http://arxiv.org/pdf/1607.04606v2.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L. and Stoyanov, V. (2020), “Unsupervised Cross-lingual Representation Learning at Scale”, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, pp. 8440–8451.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S.R., Schwenk, H. and Stoyanov, V. (2018), “XNLI: Evaluating Cross-lingual Sentence Representations”, in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (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, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186.

Geoffrey Hinton, Oriol Vinyals and Jeff Dean (2015), Distilling the Knowledge in a Neural Network, available at: https://arxiv.org/pdf/1503.02531.pdf.

Goldberg, Y. (2017), Neural Network Methods in Natural Language Processing, Morgan and Claypool Publishers.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A. and Mikolov, T. (2018), “Learning Word Vectors for 157 Languages”, in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan.

Kusner, M.J., Sun, Y., Kolkin, N.I. and Weinberger, K.Q. (2015), “From Word Embeddings to Document Distances”, in Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, JMLR.org, pp. 957–966.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), Efficient Estimation of Word Representations in Vector Space, available at: http://arxiv.org/pdf/1301.3781v3.

Reimers, N. and Gurevych, I. (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), Association for Computational Linguistics, Hong Kong, China, pp. 3982–3992.

Reimers, N. and Gurevych, I. (2020), “Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation”, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, pp. 4512–4525.

Roy, U., Constant, N., Al-Rfou, R., Barua, A., Phillips, A., & Yang, Y. (2020). LAReQA: Language-Agnostic Answer Retrieval from a Multilingual Pool. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 5919–5930. https://doi.org/10.18653/v1/2020.emnlp-main.477.

Sourav Dutta (2021), ““Alignment is All You Need”: Analyzing Cross-Lingual Text Similarity for Domain-Specific Applications”.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017), “Attention is All You Need”, in Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc, Red Hook, NY, USA, pp. 6000–6010.

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, available at: https://arxiv.org/pdf/1907.11692v1.pdf.

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press

https://doi.org/10.1017/pds.2022.153 Published online by Cambridge University Press