Building Machine Translation Tools for Patent Language: A Data Generation Strategy at the European Patent Office

Matthias Wirth[1*], Volker D. Hähnke[1], Franco Mascia[1], Arnaud Wéry[1], Konrad Vowinckel[1], Marco del Rey[2], Raúl Mohedano del Pozo[2], Pau Montes[2], Alexander Klenner-Bajaja[1]

[1] European Patent Office
Patentlaan 2
2288EE Rijswijk, Netherlands
[*] mwirth@epo.org

[2] European Patent Office
Bob-van-Benthem-Platz 1
80496 München, Germany

Abstract

The European Patent Office (EPO) is an international organisation responsible for granting patents and promoting global co-operation in the intellectual property world. With three official languages (English, German, French) and a need to constantly access and manipulate information in multiple languages, machine translation is essential for the EPO. Over the last years we have developed internal machine translation engines, specifically for the translation of patent language. This article presents our data generation strategy: it describes our approach to the generation of parallel corpora of documents, training datasets of aligned sentences, and respective evaluation datasets. Details on the challenges and technical implementation are presented, as well as statistics of the training dataset generation process.

1 Introduction and Background

The mission of the European Patent Office (EPO) is "to deliver high-quality patents and efficient services that foster innovation, competitiveness and economic growth." (European Patent Office, 2023a).

The EPO is an international organisation with English, French and German as official languages (Article 14(1) of the European Patent Convention, (EPC)) and, as a global player, it develops and promotes international cooperation at a worldwide level with organisations both inside and outside of the patent system (European Patent Office, 2023b). Both its role as a patent granting authority and being a global stakeholder in the intellectual property world requires constant access, exchange and manipulation of information in a myriad of different languages, making machine translation an indispensable tool.

Not surprisingly, the most significant part of the machine translations performed concerns the translation of patent documents.

A patent is a technical and legal document that gives inventors for a time-limited period the right to prevent others from creating, using, or selling their invention without their permission in the countries for which the patent has been granted. The basic legal requirements for a patent to be granted are, that the claimed invention is considered to be new and that it involves an inventive step in view of the state-of-the-art. According to the EPC, "the state-of-the-art shall be held to comprise everything made available to the public by means of a written or oral description, by use, or in any other way, before the date of filing of the European patent application" (Article 54(2) of the EPC). As will be appreciated, this definition imposes no restriction on language, i.e. in order to assess the basic requirements of patentability, examiners need to be able to access information in any possible language.

However, this is not the only use case for machine translation of patent documents. In the last years, the EPO has invested heavily in the development of AI-based tools for improving the efficiency of the search process by providing the best possible set of documents to start the search for an invention (Andlauer, 2018), or by automatically classifying patent documents according to the

Cooperative Patent Classification (CPC)[1]. These tools rely on language models such as BERT (Devlin et al., 2018) that our team trained from scratch on a corpus of patent text in English, thus requiring the translation of all incoming applications into English.

The EPO has a duty of confidentiality regarding unpublished applications, which makes the use of external translation providers difficult for these cases. Furthermore, patents are written using peculiar syntactic structures and employ specific terminologies, creating a hurdle for off-the-shelf translation engines trained on generic text corpora.

As part of its Strategic Plan 2023, the EPO has hence dedicated a substantial effort to the development of machine translation tools, particularly focusing on the translation of patent language. In this article we present the strategy followed to create training and evaluation datasets for the training of our own neural machine translation models for the following languages, paired to English (EN): German (DE), French (FR), Italian (IT), Dutch (NL), Spanish (ES), Chinese (ZH), Japanese (JA), Korean (KO) and Russian (RU). These languages have been selected to cover 99% of the full-text patent documents in our internal document collections.

2 Identification of Paired Documents

In order to generate a parallel corpus for training neural machine translation (NMT) models on patent language, we rely on the concept of patent family. A patent family is a collection of patent applications (or granted patents) covering the same technical content.

Patents are national legal rights, providing protection in a specific jurisdiction, e.g. a certain country. Protection in different jurisdictions requires thus filing and patent prosecution in every one of them. However, as mentioned in the previous section, the date of filing of a patent application is decisive for the assessment of the novelty and inventiveness of the claimed invention. In order to simplify the process of protecting inventions in different countries, a series of international treaties (e.g. Paris Convention, or Patent Cooperation Treaty) have been established, which among others, allow to use the filing date of the first filing (priority) for the assessment of patentability in all jurisdictions.

The generation of our parallel corpus assumes that the text of patent applications or granted patents for the same invention in different jurisdictions is likely to be a human translation of the first filing. This is a reasonable assumption, since the basic principles of patentability are common to most national or regional patent laws. Consequently, the text contents of two family members in different languages, e.g. a patent application in Germany, in German, and a patent application in the US, in English, will largely overlap, i.e. comprise the same sentences in German and English, respectively.

Additionally, certain legal provisions require the human translation of a patent publication, e.g. Article 65(1) of the EPC confers member states the right to request a translation of the patent as granted into one of its official languages.

Based on these principles, a database of parallel corpora of documents for different language pairs has been created, in which pairs of documents are stored, one document being assumed to be a human translation of the other (Täger, 2011).

3 Identification of Paired Sentences

We have seen how the concept of patent family is used to generate a parallel corpus of documents. However, we can only assume that the text contents of a pair will highly overlap. In general, it cannot be expected that sentences correspond to each other directly and in perfect order. This is why we employ a sentence alignment algorithm that identifies the sentence pairs that correspond between parallel documents. To do so, we chose the recently published *vecalign* (Thompson and Koehn, 2020) because it does not require the availability of a (however rudimentary) initial translation engine as other methods do (Sennrich and Volk, 2010). Instead, it relies on sentence embeddings, dense semantic vectors, that are generated by a multilingual pre-trained language model. These are used to assess the similarity of parallel sentences. We parameterise *vecalign* to generate alignments with a maximum sentence count of 2, allowing at maximum 1:1 sentence alignments, because this is the data we use for the training of our translation models.

[1] The CPC is a patent classification system, which has been jointly developed by the EPO and the United States Patent and Trademark Office (USPTO): https://www.cooperativepatentclassification.org/.

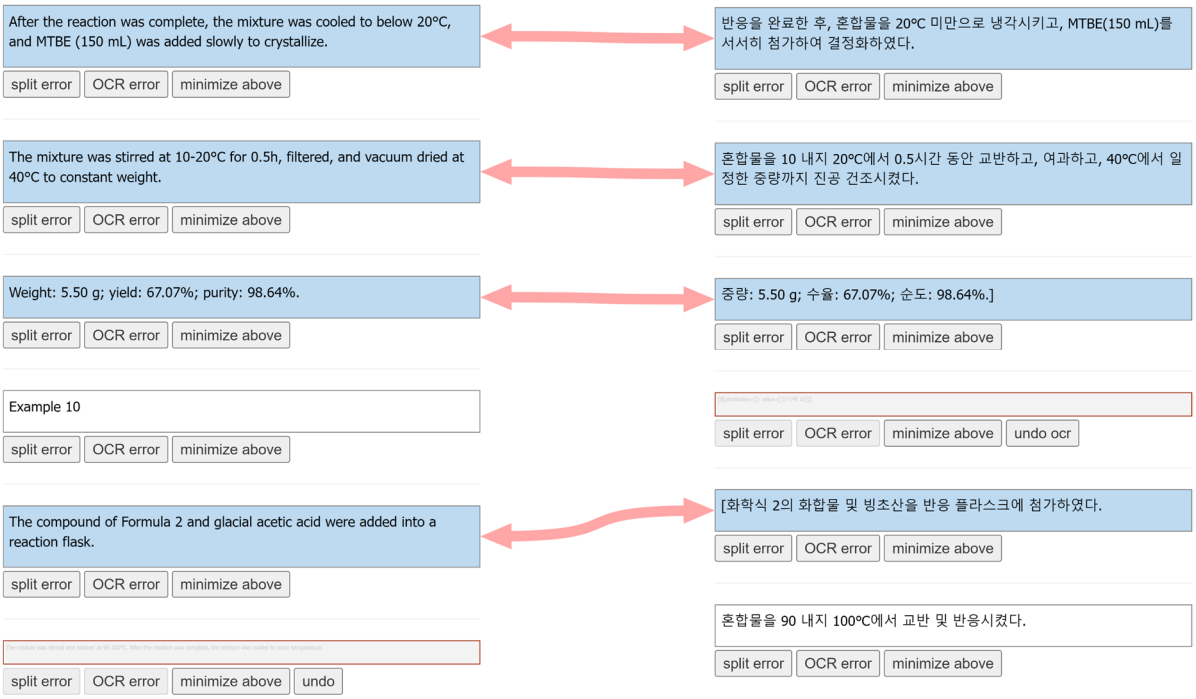

Figure 1. Screenshot of the manual sentence alignment GUI. Sentences can be mapped by clicking left and right boxes which will be visually connected by a bi-arrow representation. Buttons underneath the text boxes can be used to flag faulty text (OCR issues), sentence splitting, or to phase out already processed or irrelevant text boxes. All activities are stored on submission of the section.

A huge benefit of working in an international organisation like the EPO is that there is a high likelihood to identify a native speaker in the organization with a technical background for any of our languages of interest. To assess the alignment quality of 1:1 alignments created by *vecalign*, we compiled evaluation data sets that were used by internally recruited language experts to align sentences from parallel documents manually. Ideally, *vecalign* would confirm these 1:1 alignments.

As the only pre-processing, *vecalign* requires that documents are already split into sentences. We use different sentence splitters for different languages: the *sentence-splitter* package (*sentence-splitter*, 2023) is used for languages DE, EN, ES, FR, IT, NL, RU, the *pySBD* package (Sadvilkar and Neumann, 2020) is used for languages JA, ZH, and the *KSS* package (KSS, 2023) is used for KO. For each language, the generation of data for the manual alignment starts with a large set of paired publication sections (e.g. Description or Claims of a patent publication). For these sections, suitable pairs are selected by:

- Retaining only section pairs that have unique 1:1 assignments (one-to-many assignments occur in both directions).
- Retrieving text for each section.
- Eliminating all section pairs where at least one section has no content.
- Running the retrieved text of all sections through language detection with the *pycld3* package (pycld3, 2023); subsequently eliminating all pairs where at least one section in the pair had a disagreement in the annotated language and the *pycld3* detected language.
- Sentence splitting on all sections; subsequently eliminating all cases where percentage difference in sentence count is below 75%; subsequently eliminating all pairs where at least one section has sentence count below 10 or above 350.

The remaining section pairs were subject to further selection criteria aimed at spreading the examples uniformly over different technical fields using CPC classification. Target was 75 example section pairs per section (A-H in the CPC classification scheme); except for rare cases, this target was achieved.

The selected sections were prepared for the human alignment task by splitting them into chunks of target size 50 sentences. This was done to reduce the mental load of the cross-lingual alignment task. Reference section for the number of chunks was the English section: the number of chunks

was determined by dividing the number of available sentences by the target size and rounding the result. The sentences of the parallel non-English document were divided into the same number of chunks. The chunks were brought into an order that alternated examples from different technical fields.

The chunks were presented via a graphical user interface (Figure 1) to our internal language experts. In our sentence alignment tool, annotators can map sentences from parallel chunks to each other if they are literal translations. Additionally, they can annotate OCR and splitting errors. These examples were used to fine-tune the language specific sentence splitting, or to improve our internal text quality assessment tools.

The manually aligned sentences were used as reference for the evaluation of *vecalign*. Parallel chunks were aligned with *vecalign*, and the quality of the generated 1:1 alignments was scored with precision, recall and the $F_{0.5}$ score, weighing precision higher as recall. We chose that weighted score over the typical F_1 score because precision is our primary concern, as it measures how many (in)correct alignments *vecalign* created.

Only the fastest parameterisation of *vecalign* with maximum alignment size 2 was evaluated on all languages. This ignores the ability of *vecalign* to create many-to-one alignments in both directions. We observed in early evaluations that with higher maximum alignment sizes, recall decreases and precision increases slightly (both for 1:1 alignments). Example: for DE, with maximum alignment sizes 2, 3, 4, 5, recall develops as 0.98, 0.95, 0.95, 0.96, precision develops as 0.86, 0.93, 0.93, 0.93. Even though precision slightly increases, processing time on average doubles, which on our corpus of 1.4 billion sentence pairs makes a difference of weeks in computing time. That is why we opted for the fastest parameterization.

In *vecalign*, the semantic relatedness of text chunks is assessed based on dense vector representations generated by multi-lingual language models. The original version of *vecalign* used Language-Agnostic Sentence Representations (LASER) (Schwenk and Douze, 2017; Artetxe and Schwenk, 2019). We made use of these embeddings in the evaluations of languages DE, FR, IT, JA, NL, and ZH. Later in the project, we also evaluated Language-agnostic BERT Sentence Embedding (LaBSE) (Feng et al., 2022) for generating embeddings and found that it required only 75% of the processing time while keeping the same performance. Additionally, it is much easier to use and maintain. This is why for the last languages in this project (ES, KO, RU) we switched to LaBSE. The work in this project was structured in a linear fashion that did not allow us to go back to the first group of languages that were initially processed with *vecalign* and LASER and process them again using LaBSE. If this should be possible in the future, we will make this switch also for them.

To combat the lower precision with maximum alignment size 2, and to be able to create even higher quality aligned data, we trained a machine learning model for each language that classifies a 1:1 alignment as generated by *vecalign* as 'good' or 'bad'. This is necessary even though *vecalign* produces something like a quality indicator, the alignment cost (the higher the cost, the worse the alignment).

In Figure 2 we show the distributions of alignment cost scores of (not) manually confirmed *vecalign* 1:1 alignments for the DE–EN data set; both types overlap at almost all alignment costs. Each alignment cost score was evaluated as a possible threshold to separate good and bad alignments. The best $F_{0.5}$ score of 0.93 was observed with threshold 0.503; the best machine learning model has $F_{0.5}$ of 0.95. Observed differences between one-dimensional thresholding and machine learning are more pronounced for languages where initial *vecalign* performance is lower. The machine learning models were trained as follows:

The following features were used: (1) *vecalign* cost; (2) source sentence length (SRC); (3) target sentence length (TGT); (4) difference sentence

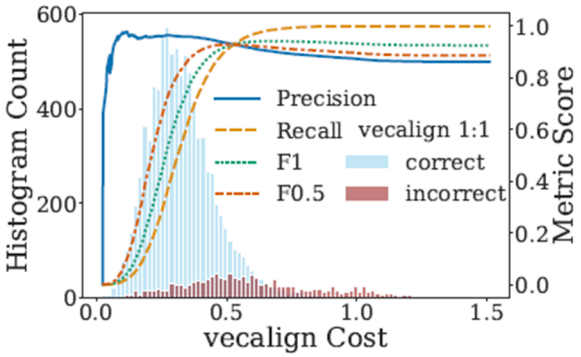

Figure 2. Distribution of alignment cost scores for *vecalign* 1:1 alignments for the DE – EN language pair. Separate plots for alignments that were (not) confirmed by manual alignment. Using alignment cost as threshold to separate confirmed/not confirmed alignment results in classification performance as indicated by precision, recall, F_1, and $F_{0.5}$.

SRC Lang.	Sent. SRC	Sent. EN	1:1 AI Manual	1:1 AI *vecalign*	1:1 AI Overlap	Recall *vecalign*	Precision *vecalign*	$F_{0.5}$ *vecalign*	$F_{0.5}$ ML
DE	12,408	11,801	9,153	10,445	9,004	0.984	0.862	0.884	0.951+
FR	10,293	10,594	8,758	9,558	8,686	0.992	0.909	0.924	0.957+
IT	19,435	20,506	16,035	17,928	15,819	0.988	0.882	0.901	0.963+
NL	4,691	4,517	3,299	3,937	3,244	0.983	0.824	0.852	0.947*
ES	6,933	6,346	4,595	5,809	4,460	0.971	0.768	0.801	0.932*
ZH	13,743	13,133	9,500	11,756	9,094	0.957	0.774	0.804	0.931*
JA	8,571	8,254	5,170	7,259	4,787	0.926	0.659	0.700	0.880*
KO	4,942	5,301	3,701	4,471	3,623	0.979	0.810	0.839	0.931*
RU	5,910	4,930	3,579	4,519	3,501	0.978	0.775	0.808	0.931*

Table 1. Evaluation statistics of *vecalign* on manually aligned reference data. The last column represents performance in the data classified as "good" alignments by the respective alignment quality model. In column "$F_{0.5}$ ML", the model type is provided with + for extremely randomised trees, and * for random forest.

length (SRC–TGT); (5) SRC character count; (6) TGT character count; (7) difference character count (SRC–TGT); (8) LaBSE cosine similarity between sentences (only for languages ES, KO, RU).

Sentence length is measured as whitespace-separated tokens, TGT language is always EN. All classification models were trained in *scikit-learn* (Pedregosa et al., 2011). Four different learning paradigms were selected for comparison: Linear Support Vector Machine, Logistic Regression, K-Nearest Neighbors, Tree-based models (Random Forest, ExtRa Trees).

For all models except the tree-based learners, a scaling model was trained on the training data and applied to the test data. The available data was split into train/test 70%/30% stratified on the target value confirmed (or not). All classifiers were evaluated in multiple configurations in a grid search, making use of a 5-fold cross validation. The tree-based model outperformed all other classifiers on all language pairs and was chosen for our data generation pipeline. The final performance is reported in Table 1.

Once the classifiers were trained, the document pairs stored in our parallel corpora database were processed using an ETL pipeline to extract sentence pairs and to store them in a PostgreSQL database, termed Sentence-Aligned Corpora Repository (SACR), ready to be used to generate training datasets.

4 Training Dataset Generation

4.1 Sources of Aligned Sentences

The starting point for the generation of the training datasets is the parallel corpus of aligned sentences stored in SACR.

For some languages, namely Italian and Dutch, for which the number of aligned sentences available was lower than 15 million datapoints, the training set was supplemented with out-of-domain data from Europarl, DGT, TED2020, EUbookshop, and the TildeMODEL datasets from OPUS (Tiedemann, 2012). Aligned sentence pairs from SACR and OPUS went through different pre-processing and filtering pipelines (described in sections 4.2 to 4.4) to end up in a pool of high-quality candidates from which training and stratified test datasets were extracted.

Datasets from external sources (OPUS) were sampled to provide a lower number of sentence pairs than those available from SACR for a given language pair, to ensure that the training sets contained more examples using the linguistic register and the in-domain terminology of the patent literature.

In the following, the process of extraction and hashing, pre-processing, and filtering is described.

4.2 Extraction Process and Hashing

Pairs fetched from SACR were selected and filtered according to the following four steps: (1) the language of the pair was confirmed with a language detection model; (2) sentences with low alignment probability were discarded; (3) sentences that were predicted to contain OCR errors were also discarded; and (4) the sentence

pairs were hashed and compared against the pairs in the Global Evaluation Dataset (GED, described in section 5) and discarded in case of a positive match.

The language of the sentences was predicted with the *fastText* model (Joulin et al., 2016). Sentences were discarded when their language was not confirmed by the model with a confidence greater than 0.8. Sentence pairs were also discarded when the alignment probability from the classification model was lower than 0.5. Furthermore, sentence pairs originating from SACR might present OCR issues that were detected with a language-agnostic heuristic based on the assumption that misspelled words are rare occurrences, i.e. they have a small edit distance from similar words that appear more often in the corpus. The sentences with a low OCR score were discarded.

All sentences from SACR and other sources were then hashed and their hashes were used to further exclude pairs that were present in the GED.

Additionally, several language-specific hash functions providing a softer match between sentences were used so that sentences such as the following should be considered to be the same: “See fig. 3 for more details.” and “see FIG 8 for more details;”. This allows for discarding sentences that are too similar in the training set and avoid having similar sentences in the training set and the GED. The sentences were normalised with language-specific rules and then hashed with SHA-256 (NIST, 2015). Among the language-specific rules, there was the lowercasing of all words in the sentence, the removal of all numbers, the removal of all white space and punctuation. Sentences as “See fig. 3 for more details.” and “see FIG 8 for more details;” would be normalised as “seefigformoredetails” before the actual hashing. Having language-specific rules instead of using a Unicode NFKD normalisation function allows to deal more precisely with orthographic variations for diacritics and ligatures. For example, the German words “verläßt” and “verlaesst” or the French “cœur” and “coeur” will be normalised and have the same hash).

4.3 Pre-processing

The data went through a series of pre-processing steps ranging from: (1) cleaning the sentences from tags and paragraph numbers, and un-escaping special characters; (2) language specific processing that can discard some sentence pairs; and (3) removal of sentence pairs after pre-processing if they are present in the GED.

The fact that a sentence pair is correctly aligned does not necessarily mean that the human translation is ideal. For example, in some cases translators will decide to leave out a comment between commas, simply because they think it does not add much information. It can also happen that for some reason a problematic pair has been aligned, for example for some language pairs, the extracted data might present encoding issues that need to be solved using heuristics, e.g. trying to reconstruct the original words or be discarded when an unambiguous correction of the data is not possible.

Other processing steps include the removal of paragraph numbers, removal of HTML tags (e.g. “<RTI>” tags) and the replacement of different escaping sequences used for Greek letters or special characters in formulas. For example, some of the sentence pairs might contain the “>” character escaped in HTML as “>”, “>” or “>” or the Greek character “• ” escaped as “&#x03B1”; “U+03B1”, “\u03B1” or even “$g(a)”.

This process was applied to sentence-pairs extracted from SACR and OPUS, and after the pre-processing step, the hashes of the data were computed again to ensure the processed sentences were not in the GED.

4.4 Filtering

After the pre-processing, several general, source-specific, and/or language-specific filters were applied to guarantee the quality of the datapoints in the training set. The following filtering steps were applied according to the source and language pair in the following order: (1) detecting whether the sentence pairs are in the wrong language; (2) detecting whether there are different numbers, symbols or brackets in the sentence pairs; (3) detecting whether there are sentences that are identical in the source and target languages; and (4) detecting whether there are duplicate pairs.

Sentence-pairs originating from OPUS sources were filtered using *fastText* and *pycld3* models to ensure they were indeed in the correct language.

Other filtering functions discarded sentence pairs in which the digits and symbols other than punctuation were different in the source and target

Lang. pair	Pairs in SACR	Discarded after extraction, filtering & processing	Pairs after extraction filtering & processing	Data from OPUS	Training set	Test set
DE-EN	210,269,198	86,131,582	124,137,616	0	124,117,544	20,072
FR-EN	63,100,060	26,922,308	36,177,752	0	36,157,742	20,010
IT-EN	8,773,195	3,199,209	5,573,986	5,503,832	11,058,292	19,526
NL-EN	16,559,613	6,081,436	10,478,177	9,565,832	20,024,215	19,794
ES-EN	77,942,615	29,511,179	48,431,436	0	48,411,478	19,958
ZH-EN	249,687,716	116,936,049	132,751,667	0	132,732,109	19,558
JA-EN	516,121,906	316,101,487	200,020,419	0	200,000,288	20,131
KO-EN	216,251,355	148,988,640	67,262,715	0	67,242,635	20,080
RU-EN	36,569,194	14,385,510	22,183,684	0	22,163,893	19,791

Table 2. Training datasets for DE, FR, IT, NL, ES, ZH, JA, KO and RU paired to EN.

sentences and in which the parentheses and brackets in the source and target sentences did not match, or were not balanced. As mentioned before these filtering functions can be adapted to take into account peculiarities of specific languages, e.g. Asian languages use different punctuation marks (e.g. "•" U+FF61 vs. "." U+002E), different number symbols (e.g. " • " U+FF11 vs. "1" U+0031), different brackets (e.g. " • " U+3010 vs. "[" U+005B) and even specific encodings for European symbol combinations (e.g. the combination "°C" U+00B0 U+0043 is written as a single-encoded character "• " U+2103). To ensure that the

translations are consistent, the symbol-matching routine must take these subtleties into account.

All sentence-pairs were further filtered using a Bloom filter and the aforementioned language-specific hash functions to detect and discard identical pairs (i.e. pairs where the sentences in the source and target language are the same) and duplicate pairs (i.e. pairs that have already been selected to be part of the training set).

As a final step, the datapoints were then divided into a training set and a test set. The test contains around 20,000 datapoints stratified into different technical fields and type of document section (Claims and Description). Stratification into technical fields was performed based on the CPC at class level[2]. All the remaining datapoints were used for the training dataset.

The process of generation of a training dataset is illustrated in Figure 3 with the example of the German–English training dataset. The resulting training datasets for all languages are described in Table 2.

5 Global Evaluation Dataset

With the purpose of measuring the performance and benchmarking the trained models, global evaluation datasets (GED) have been created for each language pair; the careful selection of sentence-pairs for each GED is aiming to ensure high-quality translations.

To generate these datasets, sentence-pairs were extracted from the SACR following the process described in the previous section. Additionally, to the extraction, pre-processing and filtering steps described in section 4, the extracted data went through the following subsequent filtering steps:

1) Text expansion/contraction filter: for each language pair, character expansion averages were calculated over the available patent corpus. The length of the target sentence was estimated using the calculated expansion average and the length of the source sentence, if the target sentence's length was outside a range of ±20% of the estimated length, the pair was discarded.
2) Bibliography exclusion: sentence pairs containing terms such as "*et al*" / "*et col*" / "*pp.*" / "*pag.*" were excluded to avoid having mixed languages in the evaluation examples (e.g. the

[2] The class level is the second level of the CPC hierarchy, it consists of 136 classes (A01 to H99, Y02, Y04 and Y10).

title of an English publication in a German source sentence).

3) LaBSE cosine similarity filter: finally, the cosine similarity between the pairs using LaBSE embeddings was used to rank the remaining pairs.

After these filtering steps, a dataset was generated by selecting sentence-pairs from the ranked list covering the following criteria:

1) Different technical fields, identified by the main CPC section (A-H) of the documents of the sentence pair - 8 in total.
2) Different sentence lengths: short, medium, long - based on the tertile distribution of sentence length in number of words (characters for Asian languages).
3) Different section types: Claims and Description.

A minimum number of sentences of 400 for each of the above combined criteria was selected, with the purpose of ensuring the statistical significance of the evaluations. The global evaluation dataset consists thus of 8×3×2×400 = 19,200 sentence-pairs per language-pair.

The hashes of the sentence-pairs in the GED were stored, so that these sentences could be excluded in the training data generation process.

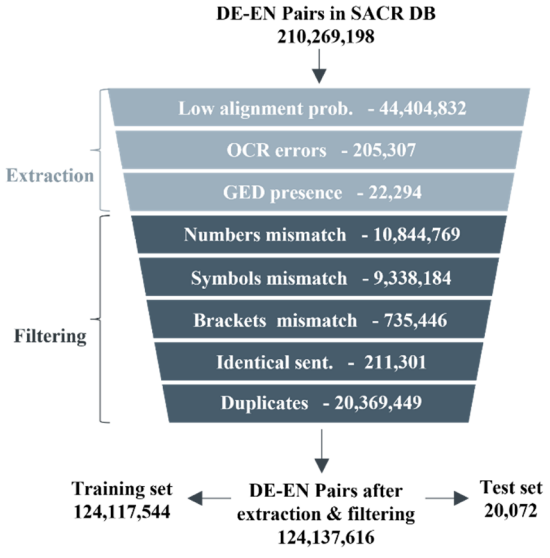

Figure 3. Example of the process of generation of the training dataset for DE-EN. For this language pair no pre-processing was required and no sentences were discarded in the filtering process due to language mismatch.

Our internally developed machine translation engines achieve the following scores: German/French–English GED BLEU (Papineni, 2002): 72.0/70.8, chrF (Popovic, 2015): 84.9/85.8 as implemented in *sacrebleu* (Post, 2018).

6 Datasets

The following datasets have been made available with this publication:

- Manual alignment data including calculated features, that were used for the training of the sentence alignment classifier, as described in section 3.
- The Global Evaluation Dataset for the language pairs French–English, and German–English.

These datasets can be found here:

https://huggingface.co/datasets/mwirth-epo/epo-nmt-datasets.

7 Conclusion

This publication outlines our strategy for the creation of parallel datasets for the training and evaluation of patent-language specific machine translation models.

First, our comprehensive approach to patent sentence alignment was detailed. We highlighted our approach to identify high quality sentence alignments from a pair of related patent documents. One major contribution of our work are the details on the development of a classification model that significantly improved precision over a *vecalign*-only-based alignment strategy. Both the evaluation of the performance of *vecalign* and the training of the subsequent classifiers relied on a set of manually curated sentences pairs created by in-house language experts, assisted by a visual interface developed in-house. The curated datasets are shared via a *huggingface* dataset repository.

In the second part of the publication, the aligned sentence corpus created from confirmed sentence pairs was described, with emphasis on the different actions taken to ensure a desired level of sentence quality and technical field balance. Details on the corpus were presented along with our approach of creating global evaluation datasets for each language pair. Our GEDs for the language pairs German–English, and French–English are shared with this publication.

It is our hope that this contribution provides a helpful insight for the interested reader into the motivations behind the efforts of the EPO regarding the development of internal machine translation engines, and how the challenge of training and evaluation data creation is being addressed.

Detailed information on the training procedure, experiments, implementation, and quality assessments of our internal machine translation engines will be the scope of a separate article.

In closing, we would like to emphasize that patents and their technical field-based classification scheme represent valuable multi-lingual resources, not only for the development of machine translation engines, but also other language processing applications.

Acknowledgements

We express our sincere gratitude to our colleagues for their efforts in various language-support tasks: Ilse Wiame, Giovanni Tommaseo, Triantafyllos Artikis, Yurika Oshino, Tobias Lüddemann, Yonghe Liu, Jie Hou, Yan Tang, Mingliu Du, Gintautas Abrasonis, Dainius Perednis, Eriks Kalejs, Peteris Skorovs, Natalia Chevtchik, Eugen Lutoschkin, and Jun-Young Bae.

References

Artetxe, Mikel and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Andlauer, D. 2018. Automatic Pre-Search: An overview. *World Patent Information*, 54:59-65.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*, 1-16.

European Patent Office, 2023a, https://www.epo.org/about-us/office/mision.html.

European Patent Office, 2023b, https://www.epo.org/about-us/services-and-activities/international-european-cooperation.html.

Feng, Fangxiaoyu, Yinfei Yang, Daniel Cer, Naveen Arivazhagan and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 1:878-891.

Joulin, Armand, Edouard Grave, Piotr Bojanowski and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv*, 1-15.

NIST 2015. Secure Hash Standard (SHS), https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

KSS (2023, 03 01). https://github.com/hyuwoongko/kss.

Papineni, Kishore, Salim Roukos, Todd Ward, Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the ACL*, 311-318.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825-2830.

Popovic, Maja. 2015. chrF: character n-gram F-score for automatic MT evaluation. *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 392-395.

Post, Matt. 2018. A Call for Clarity in Reporting BLEU Scores. *Proceedings of the Third Conference on Machine Translation: Research Papers*, 186-191.

pycld3. (2023, 03 03). https://github.com/bsolmon1124/pycld3.

Sadvilkar, Nipun and Mark Neumann. 2020. PySBD: Pragmatic Sentence Boundary Disambiguation. *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, 110-114.

Schwenk, Holger and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 157-167.

Sennrich, Rico and Martin Volk. 2010. MT-based Sentence Alignment for OCR-generated Parallel Texts. *Proceedings of AMTA 2010*.

sentence-splitter (2023, 03 01). https://github.com/mediacloud/sentence-splitter.

Täger, Wolfgang. 2011. The Sentence-Aligned European Patent Corpus. *Proceedings of the 15th Annual conference of the EAMT*, Leuven, Belgium.

Thompson, Brian and Philipp Koehn. 2020. Exploiting Sentence Order in Document Alignment. *Proceedings of the 2020 Conference on EMNLP*, 5997-6007.

Tiedemann, Jörg. 2012. Parallel Data, Tools and Interfaces in OPUS. *Proceedings of the 8th International Conference on Language Resources and Evaluation*, 2214-2218.