Natural Language Processing (2026), 1–20
doi:10.1017/nlp.2026.10021

ARTICLE

Smart bilingual focused crawling of parallel documents

Cristian García-Romero, Miquel Esplà-Gomis and Felipe Sánchez-Martínez

Dep. de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, Sant Vicent del Raspeig, Spain
Corresponding author: Cristian García-Romero; Email: cgarcia@dlsi.ua.es

(Received 15 July 2025; revised 30 December 2025; accepted 5 March 2026)

Abstract

Crawling parallel texts—texts that are mutual translations—from the Internet is usually done following a brute-force approach: documents are massively downloaded in an unguided process, and only a fraction of them end up leading to actual parallel content. In this work, we propose a smart crawling method that guides the crawl towards finding parallel content more rapidly. We follow a neural approach that consists in adapting a pre-trained multilingual language model based on the encoder of the Transformer architecture by fine-tuning it for two new tasks: inferring the language of a document from its Uniform Resource Locator (URL) and inferring whether a pair of URLs link to parallel documents. We evaluate both models in isolation and their integration into a crawling tool. The results demonstrate the individual effectiveness of both models and highlight that their combination enables us to address a practical engineering challenge: the early discovery of parallel content during web crawling in a given language pair. This leads to a reduction in the amount of downloaded documents deemed useless and yields a greater quantity of parallel documents compared to conventional crawling approaches.

Keywords: Bilingual focused crawling; language identification; parallelness inference; URL-based method

1. Introduction

Human language technologies have experienced unprecedented progress thanks to the advances in artificial intelligence and the availability of large amounts of data (Deng and Liu 2018). The availability of large text corpora is especially relevant in the field of machine translation where the state-of-the-art approach to neural machine translation (Vaswani *et al.* 2017) requires large amounts of *parallel texts*, that is texts in one language and their translation into another language. Parallel texts have also proven useful to build pre-trained language models with cross-lingual capabilities (Conneau *et al.* 2020; Kale *et al.* 2021; Reid and Artetxe 2022) and in translation-memory tools (Bowker, 2002) to assist professional translators. The reduced availability of parallel documents, particularly for low-resource language pairs, is fueling a growing interest in web mining, which has allowed to build some of the largest parallel corpora to date (El-Kishky *et al.* 2020; Bañón *et al.* 2020; Schwenk *et al.* 2021; Bañón *et al.* 2022).

State-of-the-art tools for harvesting parallel data from the Internet, like Bitextor (Bañón *et al.* 2020; Esplà-Gomis *et al.* 2016) and ILSP-FocusedCrawler (Papavassiliou, Prokopidis, and Piperidis 2018), use a web crawler to automatically browse the web and collect textual data. Web crawlers start with a list of seed URLs. The corresponding documents are downloaded and parsed, and any new URLs linked from them are added to a list of pending downloads. Crawling ends when the process is interrupted or when the list of pending downloads is exhausted. Finally, downloaded documents are processed to identify parallel content. This process is extremely inefficient, as many downloaded documents end up being discarded, either because they are not in a language of interest or because they are not parallel (Bañón *et al.* 2020). Furthermore, in large-scale crawls, where some websites are only partially crawled, this unguided approach leads to the omission of a portion of the available parallel content. This situation is exacerbated for low-resourced language pairs, as the amount of parallel documents available on the Internet is scarcer.

Focused crawling aims at addressing this problem by prioritizing those documents that are more likely to be useful for the selected downstream task (Chakrabarti, van den Berg and Dom, 1999). The literature includes several works that rely on the extracted content (Agarwal and Sureka 2014), as well as works that use only the extracted URLs from downloaded documents to prioritize crawling (Baykan, Henzinger, and Weber 2013; Hernández *et al.* 2016; Han, Wuillemin, and Senellart 2018). For the task of harvesting parallel data, we propose a smart bilingual focused crawler that ranks the URLs to be downloaded during crawling to prioritize the most promising documents, that is those that are likely to be parallel and in the desired languages. Our approach integrates the information provided by two models: one that infers the language in which a document is from its URL, and another that determines if two URLs point to two parallel documents, both without access to the documents' content. We build these models on top of XLM-RoBERTa (Conneau *et al.* 2020), a multilingual model supporting 100 languages, which we fine-tuned on publicly available datasets.

We evaluate both standalone models and their integration into a crawling tool. The language identifier is evaluated on 161 languages, the parallelness identifier on 11 language pairs, and the integration of both models into a crawling tool on 4 low-resource language pairs. The results show that both models perform their respective tasks with a reasonable accuracy, even though they only use the documents' URLs. In addition, their integration into a crawling tool leads to more parallel data with fewer documents crawled.

Our primary contribution to the state of the art is the introduction of the first focused-crawling approach designed explicitly for gathering parallel data. This approach enables the crawler to discover the parallel content in a website sooner, therefore reducing the bandwidth, and potentially the time, required for acquiring parallel corpora. Additionally, we introduce what, to the best of our knowledge, is the first approach in the literature to assess the likelihood of parallel content in documents based solely on their URLs without relying on *ad hoc* or naive rules (Dara and Lin 2016; El-Kishky *et al.* 2020).

The rest of the paper is organized as follows. Next section reviews the related work in the literature. Sections 3 and 4 then describe the two models aforementioned and evaluate them in isolation. Section 5 outlines the integration of these models into our smart bilingual focus crawling approach, which is subsequently evaluated in real crawling tasks. Finally, Section 6 ends with some concluding remarks.

2. Related work

As discussed above, conducting a general-purpose crawling for mining specific data is a waste of bandwidth and computing resources, as many downloaded documents end up being discarded for being irrelevant to the intended purpose. This challenge is addressed by developing focused crawlers (Chakrabarti, van den Berg, and Dom 1999) capable of identifying web pages relevant to the specific task at hand. This involves pruning URLs found in documents not relevant for the specific task and, optionally, ranking the URLs to prioritize downloading the most relevant ones first.

In the literature, focused crawlers are employed for various purposes, including hate detection (Agarwal and Sureka 2014), medical sentiment analysis (Abbasi *et al.* 2013), and, most prominently, topic identification (Shrivastava, Pateriya, and Kaushik 2023). Numerous URL-based approaches have been proposed to guide crawling by identifying the topic of downloaded web pages: Rajalakshmi and Aravindan (2013) use n-gram models to extract features from URLs and apply different supervised methods; Hernández *et al.* (2016) identify URL patterns given tuples of URLs and their associated topics; Han, Wuillemin, and Senellart (2018) apply reinforcement learning to prioritize downloading URLs with a higher probability of leading to documents on given topics.

Gathering parallel content from the web necessitates document alignment. However, most approaches rely on content-based methods (Buck and Koehn 2016b; Guo *et al.* 2019; Thompson and Koehn 2020) and cannot be used to prioritize URLs before downloading documents. Contrary to content-based methods, URL-based approaches offer resource savings if the necessary information can be derived from URL elements. Chen and Nie (2000) and Resnik and Smith (2003) employ string substitution with language-dependent rules to guide crawling, while Zhang, Yao, and Kit (2013) identify patterns from URL pairs differing in at least one character that may lead to parallel content. Hybrid approaches combining URL and content-based features to align parallel documents have also been proposed (Esplà-Gomis and Forcada, 2010).

Closer to our work, Barbosa, Bangalore, and Rangarajan Sridhar (2011) and Baykan, Henzinger, and Weber (2013) use URL-based machine-learning models for finding bilingual websites and for language identification, respectively. Barbosa, Bangalore, and Rangarajan Sridhar (2011) detect patterns in individual URLs identifying websites that might lead to parallel documents and process the content of the documents to identify the language—unlike our approach which only uses the URLs—and, if at least one document is found for each language of interest, they consider the website bilingual. Baykan, Henzinger, and Weber (2013) use *n*-grams from URLs to identify the language of documents and guide the crawling process; however, they only support a small set of five languages, in contrast to the 161 languages we support.

3. Language identification from URLs

We approach the inference of the language of a document from its URL as a multi-class classification problem and propose a model that utilizes XLM-RoBERTa (Conneau *et al.* 2020) as a base to produce a probability distribution over the supported languages, including an *unknown* class.

Our approach works as follows. Firstly, the input URLs undergo pre-processing to remove the protocol, decode HTML entities, pre-tokenize by splitting groups of alphabetic characters, blanks, and special characters such as underscores, and to add special tokens <s> and <s> to delimit each URL. The URLs are then processed by the XLM-RoBERTa transformer (after applying its internal sub-word tokenizer) to generate the output embeddings. In a BERT fashion (Devlin *et al.* 2019), the embedding of the first token (<s>) is used to represent each URL and passed to a feed-forward layer which is connected to a softmax output layer to obtain the final probability distribution over the set of supported languages. Figure 1 illustrates this architecture.[a]

3.1 Experimental settings

Dataset. We built a dataset consisting of pairs (u, l) associating to each URL u the language l in which the linked document is written. We build on CommonCrawl[b] (CC) snapshots CC-MAIN-2023-06 and CC-MAIN-2023-14, which contain massive collections of web documents with language automatically identified using CLD2.[c] While language is not manually annotated, CLD2 shows a very high performance with both macro-precision and macro-recall above 98% (CLD2Owners, 2014). Independent studies confirm a significant macro F1 even off-the-shelf (Lui and Baldwin 2014).

[a]Code, models, and datasets are available at https://github.com/transducens/url2lang/

[b]https://commoncrawl.org/

[c]https://github.com/CLD2Owners/cld2/

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

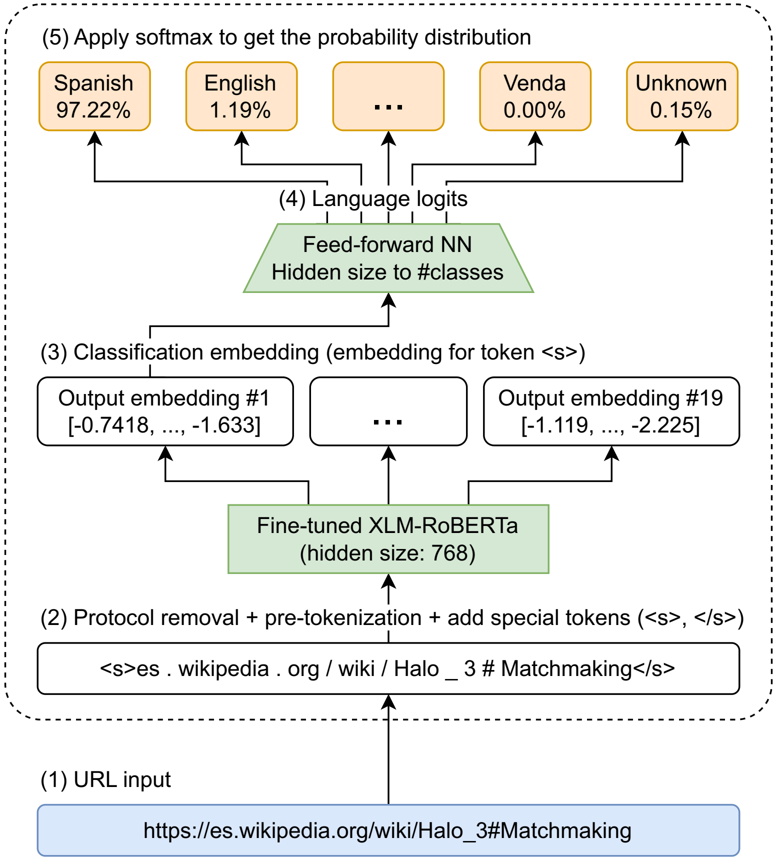

Figure 1. Architecture of the model used for language identification from URLs.

The data in CC cover 161 languages that are represented following a Zipfian distribution.[d] In order to deal with over-represented languages, a maximum of 600,000 URLs per language are collected, resulting in a corpus of about 50.4 million URLs. We split the data and use 50 million URLs for training, and the rest for development and testing (about 220,000 URLs each). We divide our data in such a way that URLs from the same web domain only appear in one of the splits. Figure 2 reports the language distribution of each set.

Evaluation metric. We present precision, recall, and F1 results, along with their aggregated counterparts: macro-precision, macro-recall, and macro-F1. These aggregated metrics represent the average values across all possible classes and offer a comprehensive overview of the systems' performance.

Baseline systems. We compare our approach to a baseline that, inspired by El-Kishky *et al.* (2020) and Baykan, Henzinger, and Weber (2013), looks at different components of the URL that may indicate the language of the linked document: subdomain, public suffix, directory names, and value of the different parameters (if any). The value of each component is compared to the ISO 639-1 and ISO 639-2 language codes.

To address potential contradictions from various URL components, we examine them in a specific order established through evaluation on the development set. The optimal sequence,

[d]https://commoncrawl.github.io/cc-crawl-statistics/plots/languages

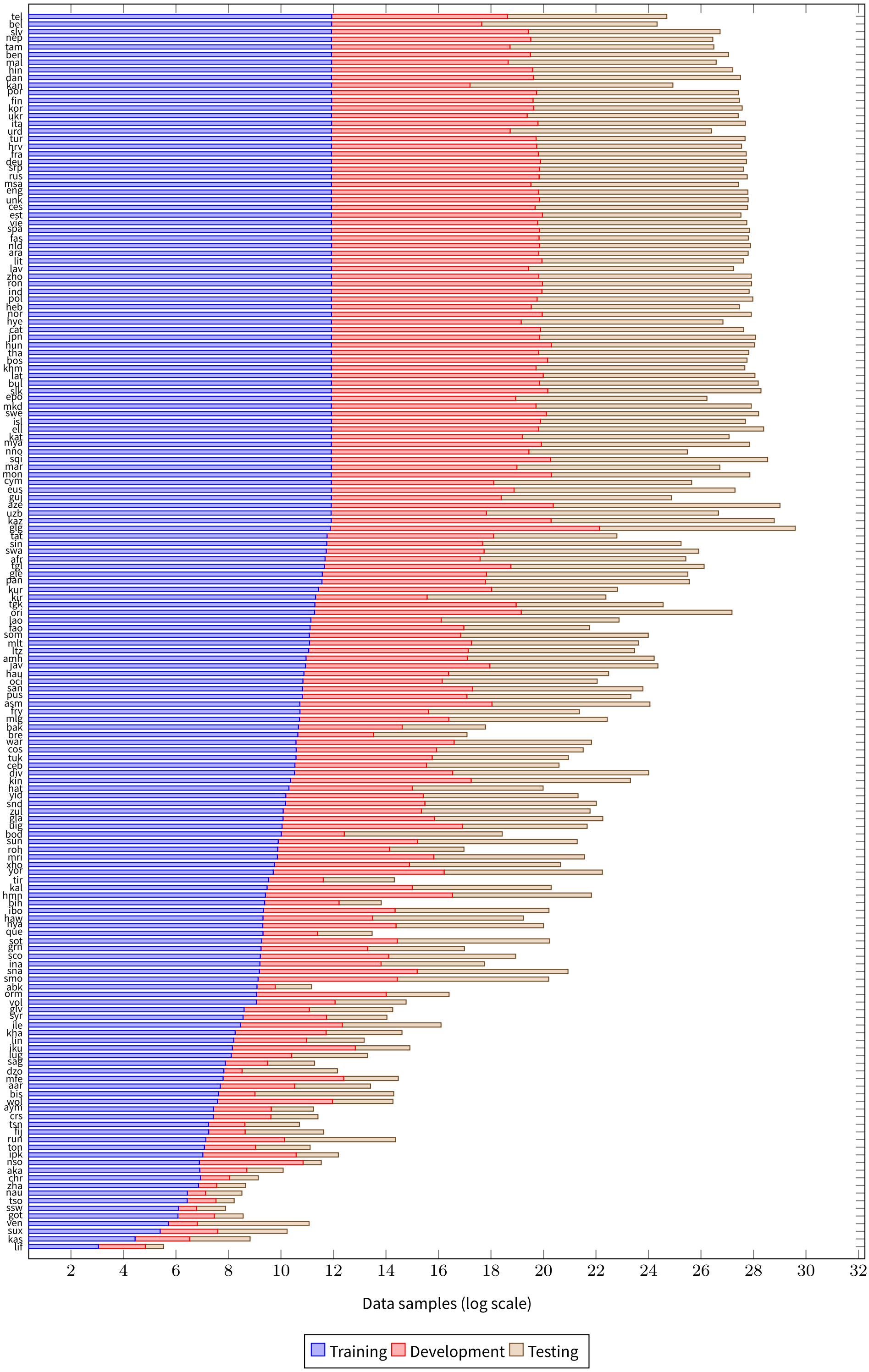

Figure 2. Language distribution for the training, development, and test sets used in the experiments for language identification.

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

Table 1. Results of the baseline, FastText, and our language identifier. Refer to Figure 3 for a breakdown per language

System	Macro Precision	Macro Recall	Macro F1
Baseline	65.29%	34.42%	38.24%
FastText	60.92%	57.32%	57.41%
Ours	**72.59%**	**69.04%**	**69.22%**

determined by assessing all possible combinations, is as follows: parameter values, directory names, public suffix, and subdomain.

Additionally, we include a baseline simpler than our transformer-based model (see Section 3) using FastText (Joulin *et al.* 2017), a common approach for language identification (Facebook AI Research, 2017; NLLB Team *et al.* 2022). This system is a linear classifier that represents text as a bag of character-level n-grams. The embeddings of these n-grams (extracted from each word independently) are averaged to form a hidden representation, which is then fed into a hierarchical softmax classifier to compute the probability distribution over the supported classes.

For this baseline, hereafter referred to as FastText, we use the same training data and data pre-processing pipeline as our approach. This ensures that both models support the same label set—enabling a direct comparison—and effectively adapts the model to classify URLs rather than text. Regarding hyperparameters, we adopt those used by the publicly available language identification model *lid.176.bin* (Facebook AI Research, 2017); namely, 16-dimensional embeddings, character n-grams from 2 to 4, learning rate of 0.1, and 10 epochs.

3.2 Training of the model

We used HuggingFace Transformers (Wolf *et al.* 2020) and PyTorch (Paszke *et al.* 2019) to fine-tune XLM-RoBERTa$_{\text{Base}}$ using the cross-entropy loss function. Based on preliminary experiments, we set the learning rate to 10^{-5} and used the AdamW optimizer (Loshchilov and Hutter, 2019) with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, $\lambda = 0.01$). We use a learning rate scheduler with an initial warm-up phase (10% of the training steps in the first epoch) where the learning rate linearly increases to the configured value and then decays following an inverse square root pattern.

Due to the extensive data in CC, utilizing the entire training set in each training epoch is impractical. Therefore, we opted to incrementally process the training set in folds. Starting with one million URLs, we add an additional million URLs after each training epoch. Training stops when further data fails to enhance the macro-F1 metric on the development set for five epochs. The training data shown in Figure 2 correspond to the data actually used for training.

Finally, following Zhang *et al.* (2021)—who argue that layers near the output of a pretrained model retain task-specific knowledge—we experimented with re-initializing the top $n \in [0, 5]$ layers of the transformer. Our findings reveal that reinitializing the top layer ($n = 1$) before fine-tuning for our downstream task slightly improves the results on the development set. Therefore, we adopted this configuration.

3.3 Results and discussion

Table 1 shows the results of the baseline system, the FastText system, and our model on the test set: our model surpasses the baseline and FastText systems across all metrics. While the baseline attains a notable macro-precision score, its macro-recall is low. This reduced recall is attributed to the absence of language information in some URLs, even in multilingual websites. This is corroborated by the significant percentage of test samples classified as *unknown* by the baseline (57.4%). The FastText system achieves better results than the baseline in terms of macro-F1 and

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

exhibits more balanced performance across the reported metrics, similar to our transformer-based model; however, it still falls behind our approach by more than 10 points in macro-precision, macro-recall, and macro-F1.

Figure 3 offers a comprehensive breakdown of the F1 metric per language, revealing the consistent superiority of our model over the baseline (left subfigure) across all languages, except for a few instances with comparable F1—notably the baseline achieves better results in Amharic and Tatar. The FastText system (right subfigure) yields results closer to our approach across languages; however, the overall tendency indicates that our model consistently outperforms it, with the only exceptions being Sanskrit, Oriya, Turkish, Hindi, Thai, Urdu, and Azerbaijani.

It is noteworthy that all the systems—baseline, FastText, and our model—yield unexpectedly low results for English; the best performing model (ours) achieves a precision of 36.6% and a recall of 58.5%. Further analysis reveals a pattern of frequent confusion between non-English URLs and English (28 languages are confused with English in at least 10% of their data). These findings align with those reported by the authors of CLD2, confirming a bias towards English, especially for Interlingua (CLD2 classifies 44.4% of these documents as English). Noticeably, our model behaves similarly and 46% of the URLs marked as Interlingua were misclassified as English.

Another source of noise in the data stems from URLs linking non-English documents, but that incorporate English text in the URL, such as the Spanish webpage https://www.dochaney.com/es/contact-us. This issue is exacerbated when the URL lacks any explicit language hint, as seen in the Spanish webpage https://hjmorelia.com/about-us. Upon manual inspection of the URLs in the test set linking to English documents, we found that 94.1% of these URLs lack any language identifier mark (e.g., "/en/" or "/∗-eng/"), compelling our model to rely on detecting the presence of English words or other intricate patterns, if any. This fact, along with the presence of webpages that link to non-English documents but contain English text in their URLs, such as the aforementioned examples of URLs linking to Spanish webpages, may be an additional reason why the model struggles with URLs linking to English documents.

The problems encountered are inherently related to the use of URLs to infer information about the content of the linked document before it is actually downloaded. URLs are often useful because they present similar semantics to the documents they link to. However, identifying the language of documents from their URLs is not always feasible due to insufficient or inaccurate semantics. During our analysis, we encountered several examples of URLs that appeared to link to a document in a particular language but instead led to a different one, as illustrated above.

4. Inferring parallelness from URLs

We model the task of determining whether two URLs point to parallel documents as a binary classification problem. Our model takes a pair of URLs as input and outputs the probability of the linked documents being parallel.

The approach mirrors that outlined in Section 3. Differences lie in the architecture: the model takes two URLs as input, and the output embedding for the first token is projected through a feed-forward layer to a single output neuron whose output is normalized using a sigmoid function, representing the probability of the two input URLs pointing to parallel documents. Both URLs undergo identical pre-processing and are separated by the special token <s>. Figure 4 illustrates this architecture.[e]

4.1 Experimental settings

Dataset. Our model is trained on a labeled dataset of URL pairs, indicating parallel or non-parallel relationships. The most relevant publicly available dataset for this task is the one distributed for the WMT16 bilingual document alignment shared task (Buck and Koehn 2016a) (hereafter WMT16);

[e]Code, models, and datasets are available at https://github.com/transducens/parallel-urls-classifier/

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

Figure 3. Language identification results on a per-language basis, comparing our model with the baseline (left subfigure) and with a FastText model trained on the same data (right subfigure). Only languages with a minimum of 100 URLs and 10 different web domains in the test set are included.

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

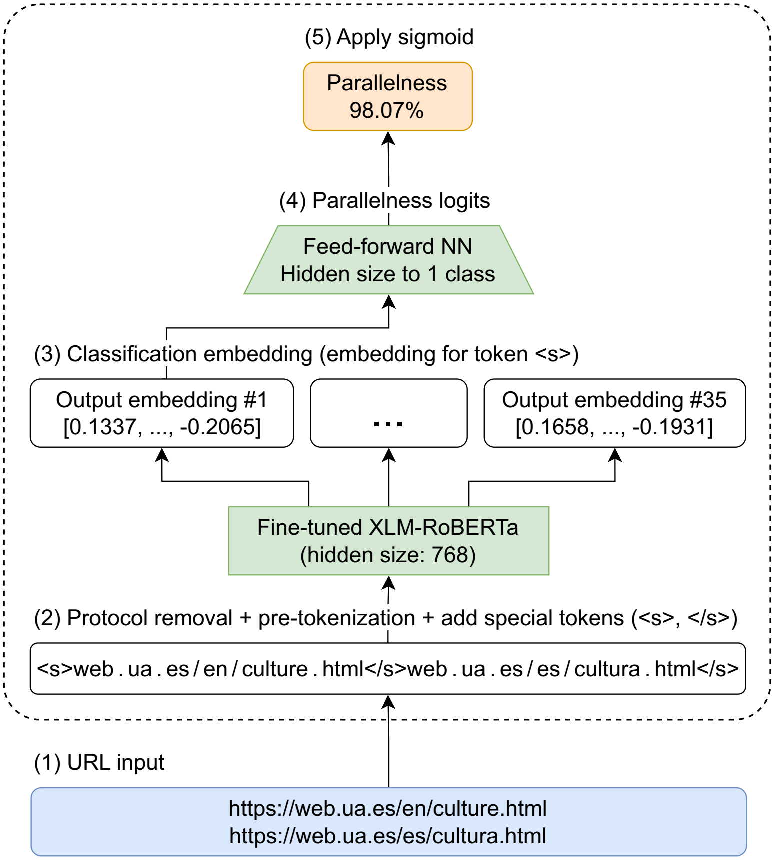

Figure 4. Architecture of the model used for inferring parallelness from URL pairs.

it consists of a collection of documents in English and French obtained from 252 multilingual websites. It is divided into training (documents from 49 websites) and test (documents from 203 websites) sets. For every document, plain text and HTML content are available, along with its URL and automatically detected language. A gold standard consisting of 1,624 URL pairs for training and 2,402 for testing is also distributed. This dataset has three significant limitations: (i) it supports only one language pair; (ii) the gold standard is relatively small and includes only a subset of the actual parallel documents in the training and test sets; and (iii) the gold standard lacks negative samples, that is URL pairs labeled as non-parallel.[f]

We deal with this last limitation by extending the original WMT16 gold standard with negative samples obtained by emulating the behavior of a web crawler—which is the scenario in which we plan to use our model—as follows. For each URL in the gold standard, we extract all the URLs linked in the corresponding HTML content and discard those pointing to documents neither in English, nor in French. We then build a set of URL pairs by pairing the original URL with every linked URL. Finally, we check if one of these pairs appears in the gold standard, and if it does, we label the rest of pairs as non-parallel; otherwise, we discard all the pairs in the set.

We alleviate the other two limitations by extending our training data with the MaCoCu (Bañón *et al.* 2022) v2 corpus, which covers 10 additional language pairs; namely, English aligned with

[f]Notice that the WMT16 shared task was initially designed as a mining task, not a classification task.

Albanian, Bulgarian, Croatian, Icelandic, Macedonian, Maltese, Montenegrin, Slovenian, Serbian, and Turkish.[g] This corpus is distributed aligned both at the segment level and at the document level with URLs attached to each pair of aligned documents; although these alignments are not as accurate as those in WMT16 because documents were automatically aligned. Additionally, this corpus does not include the original HTML content, which prevents the application of the method described above for generating negative samples. To address this limitation, we devise strategies inspired by those proposed by Esplà-Gomis *et al.* (2016) and Dara and Lin (2016) to generate negative samples without accessing the HTML content.

Creation of synthetic negative samples. In what follows we evaluate various methods for creating synthetic negative samples from the URL pairs (u_A, v_B) in the MaCoCu v2 corpus, where u_A and v_B point to documents in languages A and B, respectively. These methods are:

- **Random match:** A new URL pair (u_A, v'_B) is generated by replacing v_B with v'_B, a randomly selected URL from the pair collection. This approach is rather naive, as the resulting URLs are likely to be significantly different.
- **Remove random tokens:** Both u_A and v_B are tokenized, and random tokens are removed; random tokens never include the scheme, authority, or the port.
- **Maximize Jaccard similarity:** Similarly to *random match*, a new URL pair (u_A, v'_B) is created, but in this case v'_B is chosen so that it maximizes the Jaccard similarity to v_B. This method aims to create URL pairs very similar to the original one.

These methods assume that all URLs link documents in languages A or B, which is the case for the WMT16 dataset. However, in a realistic crawling scenario, the language cannot be identified until the document is downloaded, and it is common to encounter pairs of URLs linking documents in the same language. Hence, it is crucial to include negative samples corresponding to non-parallel documents in the same language for training. We achieve this by generating synthetic monolingual negative samples where $A = B$, using pairs of identical URLs, (u_A, u_A) and (v_B, v_B), as starting points.

We evaluated all combinations of the strategies outlined above on the WMT16 dataset, for which genuine negative samples can be obtained from HTML, as discussed earlier; the comprehensive results obtained in this evaluation are available in Appendix A. The results indicate that the most effective combination comprises *random match* (both monolingual and bilingual), *maximize Jaccard similarity* (both monolingual and bilingual), and *remove random token* (bilingual only). To create a cohesive dataset for our experiments, we apply these strategies to generate synthetic negative samples from both WMT16 and MaCoCu v2.

We use 60% of the web domains in the WMT16 training split for training and the other 40% for development.[h] The web domains in MaCoCu v2 are split as follows: 80% for training, 10% for development, and 10% for testing; we made sure that URLs from the same web domain only appear in one of these splits. Table 2 reports the amount of positive and synthetic negative samples in each set per language pair.

Evaluation metrics. We assess our model on both the official WMT16 test set and the test set described in the previous paragraph. For the WMT16 test set, we employ recall and *soft recall* (Buck and Koehn 2016a), a variant of recall used in the shared task that considers near matches. Precision cannot be used due to the non-exhaustive nature of the WMT16 gold standard, rendering the actual correct URL pairs unavailable for the test set.

[g]We choose MaCocu v2 in front of other multilingual corpora crawled from the Internet such as ParaCrawl (Bañón et al., 2020) or CCAligned (El-Kishky *et al.* 2020) as it has been reported to be of higher quality (Noord *et al.* 2023).

[h]Notice that there is no development set in the WMT16 dataset.

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

Table 2. Positive/synthetic negative samples per language pair in the training, development, and test sets are used to train the model for inferring the parallelness of two documents from their URLs

Pair	Train	Dev	Test
eng-fra	0.9k/5k	0.7k/5k	2k/15k
eng-cnr	29k/394k	2k/20k	3k/34k
eng-mlt	29k/405k	0.4k/6k	9k/125k
eng-isl	30k/400k	4k/51k	9k/125k
eng-mkd	38k/518k	6k/84k	23k/315k
eng-sqi	54k/742k	8k/113k	9k/127k
eng-slv	209k/3M	18k/245k	31k/427k
eng-bul	231k/3M	32k/437k	38k/517k
eng-srp	197k/3M	57k/729k	51k/646k
eng-hrv	246k/3M	49k/669k	22k/292k
eng-tur	444k/6M	66k/879k	50k/669k

As regards the evaluation on the test set described above, the same metrics described in Section 3.1 are used: precision, recall, F1, macro-precision, macro-recall, and macro-F1.

Baseline system. We use the official baseline of the WMT16 shared task[i] (Buck and Koehn 2016a).[j] This baseline defines a set of tokens considered to be potential language identifiers in a URL; a given pair of URLs is aligned if they can be transformed into the same string by removing one or more of these tokens. We have extended this baseline to support all the languages covered in our experiments. Similar to the already supported languages, the set of tokens for additional languages includes the name of the language in English, its endonym, the ISO 639-1 and ISO 639-2 language codes, and combinations of the ISO 639-1 language codes and the ISO 3166-1 country codes where the language is official, separated by a hyphen; for example for Albanian: *albanian*, *shqip*, *sqp*, *alb*, *sq*, *sq-sq*, *sq-ks*.

4.2 Training of the model

We employed the same training configuration as regards learning rate, optimizer, hyperparameters, and loss function, as described in Section 3.2. Training stops if the macro-F1 computed on the development set does not improve after five epochs. In addition, the top layer of the XLM-RoBERTa$_{\text{Base}}$ model used is re-initialized before training.

4.3 Results and discussion

Table 3 reports the results in terms of recall and soft recall on the WMT16 test set. The figures in this table can be directly compared to those achieved by the systems submitted to the shared task (Buck and Koehn (2016a, Tables 3 & 4); 19 submissions from 11 distinct research groups. To compute them, we obtain the Cartesian product of all English and French URLs in the WMT16

[i]https://github.com/christianbuck/wmt16-document-alignment-task/

[j]None of the submissions to the WMT16 shared task can be considered as a baseline because all of them rely on the content of the documents to align them.

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

Table 3. Recall and soft recall obtained by our parallel URL identifier and the baseline on the WMT16 test set

System	Recall	Soft Recall
Baseline	59.91	59.91
Ours	**62.99**	**66.69**

Table 4. Results for the parallelness identifier from URLs on the dataset described in Section 4.1, considering all language pairs collectively

Class	Metric	Baseline	Ours
	Precision	64.37	**76.28**
Positive	Recall	12.78	**61.79**
	F1	21.33	**68.28**
	Precision	93.86	**97.19**
Negative	Recall	**99.47**	98.57
	F1	96.58	**97.87**
	Precision	79.12	**86.74**
Macro	Recall	56.13	**80.18**
	F1	58.96	**83.08**

test set and apply our model and the baseline to each URL pair. As WMT16 only allows 1-to-1 URL alignments, our model outputs the most probable alignment among those identified. The results obtained are promising: our approach not only outperforms the baseline in almost 7 points of soft recall but also four of the nineteen systems submitted to the shared task. This is especially relevant given that all the systems submitted compare the content of documents to determine whether they are parallel or not, while our model solely relies on URLs.

Table 4 presents the results on the test set described in Section 4.1 for all language pairs combined; Figure 5 displays the macro F1 score for each language pair individually. Table 4 reveals the baseline's poor performance for the positive class, with a low recall below 15%, in contrast to the high recall (above 99%) for the negative class. This suggests that the baseline's simplicity struggles with the complexities of parallel URLs. However, it achieves a competitive macro-precision, although not surpassing our system. This is because it looks for explicit language hints in the URLs and if a URL pair is deemed parallel, it is likely to be a true positive, except for potential confusion with common strings in ISO-639 codes (e.g., "isl" confused with "island" instead of Icelandic).

Figure 5 shows that our model systematically outperforms, in terms of macro-F1, the baseline in all languages, but French. Manual inspection confirmed a significant difference between the English–French test set, derived from the WMT16 dataset, and the rest of the test sets, derived from the MaCoCu v2 corpus. About 60% of the positive samples in the English–French dataset exclusively use language identification tokens in URLs as the method to identify translated documents.[k] Conversely, for the datasets used for the rest of languages this percentage ranges from 10% to 15%. The fact that the baseline designed for the WMT16 task exclusively builds on language

[k]For example, pairs of URLs www.un.org/en/ and www.un.org/fr/, that only differ in the tokens *en* and *fr*.

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

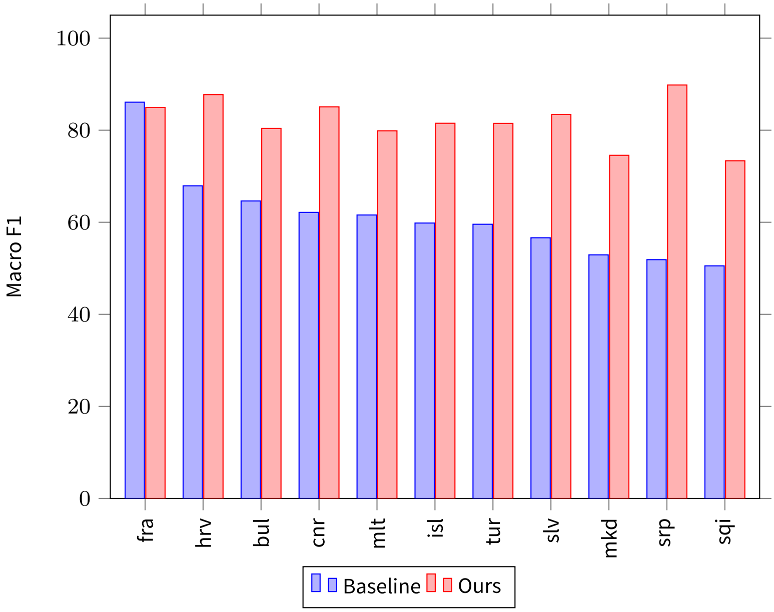

Figure 5. Macro-F1 scores per language (paired with English) for the parallelness identifier from URLs on the dataset described in Section 4.1.

identification tokens to detect parallel documents explains the high performance on this specific test set. While the details about the methodology followed to build the WMT16 gold standard are not public, it seems likely that the procedure followed somehow biased this dataset towards over-representing websites using language identification tokens.

5. Smart bilingual-focused crawling

In this section, we describe and evaluate our smart bilingual focus crawling strategy leveraging the models discussed in Sections 3 and 4. We implement our crawling approach on Heritrix 3,[l] an open-source and widely used crawler.[m] Like other crawling tools, Heritrix operates with a list of pending downloads initially populated with a set of seed URLs. During the crawl, URLs are extracted from this list, and the downloaded documents are parsed to append any newly discovered URLs to it. By default, Heritrix explores the URLs in the list of pending downloads in the order in which they were added (i.e., breadth-first search). However, we modify this behavior by using a priority queue where the priority of each URL is determined by combining the probabilities obtained from the two models described above. Additionally, we ensure that the seed URLs are always prioritized for download in the first place.

Figure 6 illustrates the general architecture of our crawling approach. For a specified pair of languages A and B, we proceed as follows:

1. A URL u is taken from the list of pending downloads and the corresponding document D_u is downloaded.

[l]https://github.com/internetarchive/heritrix3/

[m]We created a fork of Heritrix 3 to implement the required changes for our experiment. The code is available at https://github.com/cgr71ii/heritrix3/tree/puc_uri_cost/

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

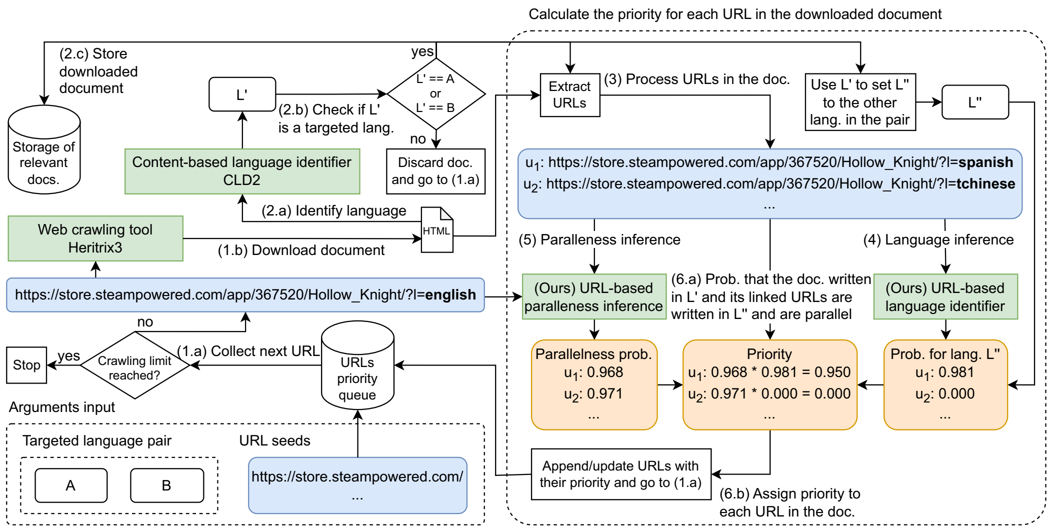

Figure 6. General architecture of our approach for smart bilingual focused crawling of parallel documents. In this example, *A* and *B* are English and Spanish, the language L' of the downloaded document is English, and the language L'' used to obtain the language probabilities is Spanish.

2. CLD2 is used to determine the language of D_u; if it is not either A or B, the document is discarded, otherwise the document is stored and the process continues.
3. D_u is parsed and the collection of URLs $\{v_i\}_1^N$ linked from it is extracted.
4. The model that identifies the language of a document from its URL is used to determine the probability of each URL v_i being in language A if D_u is in language B, or in language B if D_u is in language A.
5. The model that infers parallelness from URLs is used to obtain the probability of each pair $\{(u, v_i)\}_1^N$ of being parallel.
6. Each URL v_i is added to the list of pending downloads with a priority score determined by multiplying the probabilities obtained in steps 4 and 5. If v_i is already in the list, its priority is updated only if the new priority is higher.

Note that this process is based on the assumption that web pages containing the same content but in different languages are typically mutually linked.

5.1 Experimental settings

Dataset. We evaluate the crawling strategy described above on four language pairs: English–Icelandic (eng-isl), English–Maltese (eng-mlt), English–Finnish (eng-fin), and Spanish–Basque (spa-eus). It is worth noting that neither Finnish, Spanish, nor Basque were used to fine-tune the model for inferring parallelness from URLs, and Maltese was not even used to train the base model, XLM-RoBERTa, on which our models build.

We built a collection of seed URLs for each of the four bilingual crawling tasks. To ensure that the websites to be crawled actually contained parallel data, we used the Paracrawl v9 parallel corpus to identify productive websites for the languages covered in our experiment. Paracrawl consists of parallel sentences for 43 language pairs, along with their corresponding source URLs.

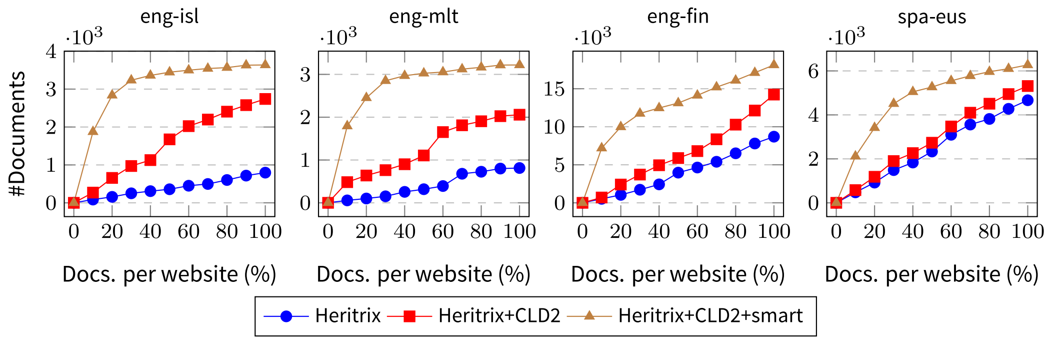

Figure 7. For the three crawlers (Heritrix, Heritrix+CLD2, and Heritrix+CLD2+smart) and the language pairs eng-isl, eng-mlt, eng-fin, and spa-eus, thousands of parallel documents retrieved (*y*-axis) as a function of the percentage of documents downloaded (*x*-axis) from each website. Each data point accumulates the number of parallel documents downloaded from each website up to the percentage indicated on the *x*-axis.

For each language pair, we built the list of unique URLs[n] and grouped them by websites; we then ranked websites by the amount of URLs in this list. Websites present in the training and development sets used to fine-tune the models described in Sections 3 and 4 were excluded. Ultimately, a list of seed URLs was created for each language pair by collecting the home page of the 200-top websites.[o]

Evaluation. We measure the performance of our approach in terms of the amount of parallel data downloaded at different moments of a crawling process. We include three models in our comparison: (a) the original implementation of Heritrix; (b) the proposed approach that integrates the two models proposed in Sections 3 and 4, along with CLD2 (Heritrix+CLD2+smart); and (c) a crawler integrating only CLD2 to exclude documents not in the targeted languages (Heritrix+CLD2). The inclusion of the last crawler helps determine the individual contributions of the proposed models and CLD2. We run each version of Heritrix on each website included in the experiment for a maximum of 48 hours. We then evaluate the amount of parallel data obtained by deciles of the total data downloaded per website.

To measure the amount of parallel data acquired, we used Bitextor[p] because manual assessment would have been prohibitively expensive. Although Bitextor may introduce some noise, this equally affects all evaluated approaches. In any case, it is important to note that Bitextor is a state-of-the-art tool for harvesting parallel content (Bañón *et al.* 2020; Esplà-Gomis *et al.* 2016) widely used and that has been utilized to create popular datasets such as ParaCrawl (Bañón *et al.* 2020) and MaCoCu (Bañón *et al.* 2022).

5.2 Results and discussion

Figure 7 illustrates the results obtained by the three models; the x-axis corresponds to the percentage[q] of the total number of documents downloaded for each website after 48 hours of crawling,

[n]Since some of the URLs in Paracrawl do not exist anymore, we discarded those with an HTTP connection status other than *OK* (HTTP code 200).

[o]The home page of a website is defined as its URL without any additional resource (e.g., *https://en.wikipedia.org* for *https://en.wikipedia.org/wiki/Deep_learning*).

[p]https://github.com/bitextor/bitextor/

[q]The reason for using relative rather than absolute values on the x-axis is that the number of documents downloaded from each website may vary due to external factors such as server workload or network speed.

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

while the y-axis represents the amount of parallel data identified at each moment. The number of parallel documents for a given percentage of downloaded data was computed separately for each website, and the figure shows the aggregated results. It is worth noting that the number of parallel documents identified can serve as a proxy for crawling speed, avoiding the drawbacks of measuring crawling times, such as server workload or computing resource constraints. If more parallel documents are identified with the same amount of downloaded documents, then we are potentially improving crawling speed and bandwidth usage with respect to the downloaded documents.

The results obtained for all language pairs confirm that the Heritrix+CLD2+smart approach is capable of identifying most of the available parallel data earlier than the other two approaches. As expected, the difference between the three approaches diminishes as a higher fraction of the available data is downloaded, and they would eventually converge if the crawling time was enough to allow crawling all the data in these websites. Another notable observation is that the trend is consistent across all language pairs, regardless of whether they were seen during training, partially confirming the zero-shot capabilities of our approach (Heritrix+CLD2+smart). Notice that while our approach for inferring parallelness is language-independent, the model used to identify the language of documents from their URLs (see Section 3) was fine-tuned on all languages included in the experiments. Hence, a more extensive evaluation on a larger number of language pairs is required to robustly confirm the zero-shot capabilities of our method.

6. Concluding remarks

In order to optimize the crawling of parallel documents from the Internet and save bandwidth, we have introduced two models that work solely on the URLs and that are then integrated into a crawling tool. The first model is able to infer the language in which a document is from its URL; the second one determines if two URLs point to two parallel documents. We have thoroughly evaluated the two models in isolation, and they have demonstrated utility for their respective tasks.

We have integrated these two models into a crawling tool to prioritize downloading URLs that are more likely to lead to parallel content in the desired languages. Experiments with four different language pairs demonstrate a clear increase in the ratio of parallel documents to downloaded documents, thereby reducing the number of discarded documents. This improves bandwidth usage and has the potential to reduce crawling time while yielding a greater quantity of parallel documents compared to regular crawling approaches. Additionally, the approach proves effective with language pairs not seen by the model used for inferring parallelness from URLs.

We have not assessed multilingual language models other than XLM-RoBERTa. Exploring newer models such as mT5 (Xue *et al.* 2021) could provide further evidence regarding the effectiveness of our approach but would require greater computational resources due to the trend of increasing the number of model parameters (Smith *et al.* 2022). Nevertheless, our main conclusion remains unaffected: URL-based models built on top of a multilingual language model can effectively guide crawling to obtain more parallel documents than traditional approaches.

All code, datasets derived from the corpora used, and models used in our experiments are made publicly available to facilitate replication and validation by other researchers, as well as for their use for smart bilingual focus crawling.

Acknowledgements. This work was supported by the Spanish Ministry of Science and Innovation, the Spanish Research Agency (AEI/10.13039/501100011033) and the European Regional Development Fund, A way to make Europe [grant number PID2021-127999NB-I00]; Cristian García-Romero is funded by Generalitat Valenciana and the European Social Fund [research grant CIACIF/2021/365]. The computational resources used were funded by the European Regional Development Fund through project IDIFEDER/2020/003.

Competing interests. The author(s) declare none.

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

References

Abbasi A., **Fu T.**, **Zeng D.** and **Adjeroh D.** (2013). Crawling credible online medical sentiments for social intelligence. In *2013 International Conference on Social Computing*. IEEE, pp. 254–263.

Agarwal S. and **Sureka A.** (2014). A focused crawler for mining hate and extremism promoting videos on youtube. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*. Santiago, Chile: Association for Computing Machinery, pp. 294–296. https://doi.org/10.1145/2631775.2631776

Bañón M., **Chen P.**, **Haddow B.**, **Heafield K.**, **Hoang H.**, **Esplà-Gomis M.**, **Forcada M.L.**, **Kamran A.**, **Kirefu F.**, **Koehn P.**, **Ortiz Rojas S.**, **Pla Sempere L.**, **Ramírez-Sánchez G.**, **Sarrías E.**, **Strelec M.**, **Thompson B.**, **Waites W.**, **Wiggins D.** and **Zaragoza J.** (2020). ParaCrawl: web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4555–4567. https://doi.org/10.18653/v1/2020.acl-main.417

Bañón M., **Esplà-Gomis M.**, **Forcada M.L.**, **García-Romero C.**, **Kuzman T.**, **Ljubešić N.**, **van Noord R.**, **Sempere L.P.**, **Ramírez-Sánchez G.**, **Rupnik P.**, **Suchomel V.**, **Toral A.**, **van der Werff T.** and **Zaragoza J.** (2022). MaCoCu: Massive collection and curation of monolingual and bilingual data: Focus on under-resourced languages. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*. Ghent, Belgium: European Association for Machine Translation, pp. 303–304. https://aclanthology.org/2022.eamt-1.41

Barbosa L., **Bangalore S.** and **Rangarajan Sridhar V.K.** (2011). Crawling back and forth: Using back and out links to locate bilingual sites. In Wang, H. and Yarowsky, D. (eds), *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, pp. 429–437. https://aclanthology.org/I11-1048

Baykan E., **Henzinger M.** and **Weber I.** (2013). A comprehensive study of techniques for URL-based web page language classification. *ACM Transactions on the Web* **7**(1), 1–37. https://dl.acm.org/doi/pdf/10.1145/2435215.2435218

Bowker L. (2002). *Computer-Aided Translation Technology: A Practical Introduction*. University of Ottawa Press.

Buck C. and **Koehn P.** (2016a). Findings of the WMT 2016 bilingual document alignment shared task. In Bojar O., Buck C., Chatterjee R., Federmann C., Guillou L., Haddow B., Huck M., et al., (eds), *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Berlin, Germany: Association for Computational Linguistics, pp. 554–563. https://doi.org/10.18653/v1/W16-2347

Buck C. and **Koehn P.** (2016b). Quick and reliable document alignment via TF/IDF-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Berlin, Germany: Association for Computational Linguistics, pp. 672–678. https://doi.org/10.18653/v1/W16-2365.

Chakrabarti S., **van den Berg M.** and **Dom B.** (1999). Focused crawling: A new approach to topic-specific Web resource discovery. *Computer Networks* **31**(11), 1623–1640. https://doi.org/10.1016/S1389-1286(99)00052-3

Chen J. and **Nie J.-Y.** (2000). Parallel web text mining for cross-language IR. In *Content-Based Multimedia Information Access*, vol. **1**. Paris, France, pp. 62–77.

CLD2Owners. (2014). *Evaluate cld2 20140122 1024k*. Available at https://github.com/CLD2Owners/cld2/blob/master/docs/evaluatecld2large20140122.txt (Accessed February 5, 2024).

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*. Online. Association for Computational Linguistics, pp. 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

Dara A.A. and **Lin Y.-C.** (2016). YODA system for WMT16 shared task: Bilingual document alignment. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Berlin, Germany: Association for Computational Linguistics, pp. 679–684. https://doi.org/10.18653/v1/W16-2366

Deng L. and **Liu Y.** (2018). *Deep Learning in Natural Language Processing*, 1st Edn. Singapore: Springer Singapore.

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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423

Esplà-Gomis M., **Forcada M.**, **Ortiz-Rojas S.** and **Ferrández-Tordera J.** (2016). Bitextor's participation in WMT'16: Shared task on document alignment. In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers. Berlin, Germany: Association for Computational Linguistics, pp. 685–691. https://doi.org/10.18653/v1/W16-2367

Esplà-Gomis M. and **Forcada M.L.** (2010). Combining content-based and URL-based heuristics to harvest aligned bitexts from multilingual sites with bitextor. *The Prague Bulletin of Mathematical Linguistics* **93**(1), 77–86.

Facebook AI Research. (2017). Language identification with fasttext. *Available at* https://fasttext.cc/docs/en/language-identification.html (Accessed December 16, 2025).

Guo M., **Yang Y.**, **Stevens K.**, **Cer D.**, **Ge H.**, **Sung Y.-H.**, **Strope B.** and **Kurzweil R.** (2019). Hierarchical document encoder for parallel corpus mining. In Bojar O., Chatterjee R., Federmann C., Fishel M., Graham Y., Haddow B., Huck M., et al., (eds), *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Florence, Italy: Association for Computational Linguistics, pp. 64–72. https://doi.org/10.18653/v1/W19-5207

Han M., **Wuillemin P.-H.** and **Senellart P.** (2018). Focused crawling through reinforcement learning. In Web Engineering: 18th International Conference (ICWE. Cáceres, Spain: Springer-Verlag, pp. 261–278. https://doi.org/10.1007/978-3-319-91662-0_20

Hernández I., **Rivero C.R.**, **Ruiz D.** and **Corchuelo R.** (2016). CALA: ClAssifying links automatically based on their URL. *Journal of Systems and Software* **115**, 130–143. https://www.sciencedirect.com/science/article/pii/S016412121600042X

Joulin A., **Grave E.**, **Bojanowski P.** and **Mikolov T.** (2017). Bag of tricks for efficient text classification. In Lapata M., Blunsom P. and Koller A. (eds), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 427–431. https://aclanthology.org/E17-2068/

Kale M., **Siddhant A.**, **Al-Rfou R.**, **Xue L.**, **Constant N.** and **Johnson M.** (2021). NmT5 - Is parallel data still relevant for pre-training massively multilingual language models? In Zong C., Xia F., Li W. and Navigli R. (eds), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics And The 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online. Association for Computational Linguistics, pp. 683–691. https://doi.org/10.18653/v1/2021.acl-short.87

El-Kishky A., **Chaudhary V.**, **Guzmán F.** and **Koehn P.** (2020). CCAligned: A massive collection of cross-lingual web-document pairs. In Webber B., Cohn T., He Y. and Liu Y. (eds), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (Emnlp)*. Online. Association for Computational Linguistics, pp. 5960–5969. https://doi.org/10.18653/v1/2020.emnlp-main.480

Loshchilov I. and **Hutter F.** (2019). Decoupled weight decay regularization. In *Proceedings of the Seventh International Conference on Learning Representations (ICLR 2019)*. New Orleans, USA, pp. 1–8.

Lui M. and **Baldwin T.** (2014). Accurate language identification of Twitter messages. In Farzindar A., Inkpen D., Gamon M. and Nagarajan M. (eds), *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 17–25. https://doi.org/10.3115/v1/W14-1303

NLLB Team, **Costa-Jussà M.R.**, **Cross J.**, **Çelebi O.**, **Elbayad M.**, **Heafield K.**, **Heffernan K.**, **Kalbassi E.**, **Lam J.**, **Licht D.**, **Maillard J.**, **Sun A.**, **Wang S.**, **Wenzek G.**, **Youngblood A.**, **Akula B.**, **Barrault L.**, **Gonzalez Mejia G.**, **Hansanti P.**, **Hoffman J.**, **Jarrett S.**, **Sadagopan K.R.**, **Rowe D.**, **Spruit S.**, **Tran C.**, **Andrews P.**, **Ayan N.F.**, **Bhosale S.**, **Edunov S.**, **Fan A.**, **Gao C.**, **Goswami V.**, **Guzmán F.**, **Koehn P.**, **Mourachko A.**, **Ropers C.**, **Saleem S.**, **Schwenk H.** and **Wang J.** (2022). No language left behind: scaling human-centered machine translation. https://doi.org/10.48550/arXiv.2207.04672 arXiv preprint arXiv:2207.04672.

van Noord R., **Chichirau M.**, **Esplà-Gomis M.**, **Kuzman T.**, **Ljubešić N.**, **Ramírez-Sánchez G.**, **Rupnik P.** and **Toral A.** (2023). Evaluation of data release 2. Technical report. MaCoCu Action. https://macocu.eu/static/media/second-report.453a82100b1ec3647012.pdf

Papavassiliou V., **Prokopidis P.** and **Piperidis S.** (2018). Discovering parallel language resources for training MT engines. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). https://aclanthology.org/L18-1599

Paszke A., **Gross S.**, **Massa F.**, **Lerer A.**, **Bradbury J.**, **Chanan G.**, **Killeen T.**, **Lin Z.**, **Gimelshein N.**, **Antiga L.**, **Desmaison A.**, **Kopf A.**, **Yang E.**, **DeVito Z.**, **Raison M.**, **Tejani A.**, **Chilamkurthy S.**, **Steiner B.**, **Fang L.**, **Bai J.** and **Chintala S.** (2019). PyTorch: An imperative style, high-performance deep learning library. In *33rd Conference on Neural Information Processing Systems (NEURIPS 2019)*. Vancouver, Canada, pp. 1–12.

Rajalakshmi R. and **Aravindan C.** (2013). Web page classification using n-gram based URL features. In *2013 Fifth International Conference on Advanced Computing (ICoAC)*. Chennai, India: IEEE, pp. 15–21.

Reid M. and **Artetxe M.** (2022). PARADISE: Exploiting parallel data for multilingual sequence-to-sequence pretraining. In Carpuat M., de Marneffe M.-C. and Meza Ruiz I.V. (eds), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, pp. 800–810. https://doi.org/10.18653/v1/2022.naacl-main.58

Resnik P. and **Smith N.A.** (2003). The web as a parallel corpus. *Computational Linguistics* **29**(3), 349–380. https://doi.org/10.1162/089120103322711578

Schwenk H., **Wenzek G.**, **Edunov S.**, **Grave E.**, **Joulin A.** and **Fan A.** (2021). CCMatrix: Mining billions of high-quality parallel sentences on the web. In Zong C., Xia F., Li W. and Navigli R. (eds), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online. Association for Computational Linguistics, pp. 6490–6500. https://doi.org/10.18653/v1/2021.acl-long.507

Shrivastava G.K., **Pateriya R.K.** and **Kaushik P.** (2023). An efficient focused crawler using LSTM-CNN based deep learning. *International Journal of System Assurance Engineering and Management* **14**(1), 391–407.

Smith S., **Patwary M.**, **Norick B.**, **LeGresley P.**, **Rajbhandari S.**, **Casper J.**, **Liu Z.**, **Prabhumoye S.**, **Zerveas G.**, **Korthikanti V.**, **Zhang E.**, **Child R.**, **Yazdani Aminabadi R.**, **Bernauer J.**, **Song X.**, **Shoeybi M.**, **He Y.**, **Houston M.**, **Tiwary S.** and **Catanzaro B.** (2022). Using DeepSpeed and Megatron to train megatron-Turing NLG 530B, a large-scale generative language model. arXiv preprint arXiv:2201.11990 (February). https://doi.org/10.48550/arXiv.2201.11990

Thompson B. and **Koehn P.** (2020). Exploiting sentence order in document alignment. In Webber B., Cohn T., He Y. and Liu Y. (eds), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online. Association for Computational Linguistics, pp. 5997–6007. https://doi.org/10.18653/v1/2020.emnlp-main.483

Vaswani A., **Shazeer N.**, **Parmar N.**, **Uszkoreit J.**, **Jones L.**, **Gomez A.N.**, **Kaiser Ł.** and **Polosukhin I.** (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach, USA, pp. 6000–6010.

Wolf T., **Debut L.**, **Sanh V.**, **Chaumond J.**, **Delangue C.**, **Moi A.**, **Cistac P.**, **Rault T.**, **Louf R.**, **Funtowicz M.**, **Davison J.**, **Shleifer S.**, **von Platen P.**, **Ma C.**, **Jernite Y.**, **Plu J.**, **Xu C.**, **Le Scao T.**, **Gugger S.**, **Drame M.**, **Lhoest Q.** and **Rush A.M.** (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online. Association for Computational Linguistics, pp. 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

Xue L., **Constant N.**, **Roberts A.**, **Kale M.**, **Al-Rfou R.**, **Siddhant A.**, **Barua A.** and **Raffel C.** (2021). MT5: A massively multilingual pre-trained text-to-text transformer. In Toutanova K., Rumshisky A., Zettlemoyer L., Hakkani-Tur D., Beltagy I., Bethard S., Cotterell R., Chakraborty T. and Zhou Y. (eds), *Proceedings of the 2021 Conference of the North American Chapter of the Association For Computational Linguistics: Human Language Technologies*. Online. Association for Computational Linguistics, pp. 483–498. https://doi.org/10.18653/v1/2021.naacl-main.41

Zhang C., **Yao X.** and **Kit C.** (2013). Finding more bilingual webpages with high credibility via link analysis. In Serge S., Pierre Z. and Reinhard R. (eds), *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 138–143. https://aclanthology.org/W13-2517

Zhang T., **Wu F.**, **Katiyar A.**, **Weinberger K.Q.** and **Artzi Y.** (2021). Revisiting few-sample bert fine-tuning. In *Proceedings of the Ninth International Conference on Learning Representations (ICLR 2021)*, pp.1–22. Virtual.

Appendix A. Synthetic negative samples for inferring parallelness from URLs

This Appendix describes the evaluation of the different strategies described in Section 7 to produce synthetic negative samples consisting of pairs of URLs corresponding to non-parallel documents. We assess all combinations of these strategies on the WMT16 dataset, for which negative samples can be straightforwardly obtained from the HTML content.

Our approach involves using only the WMT16 training split to generate negative samples using the combination of strategies under evaluation and adopting a 10-fold cross-validation strategy. In each cross-validation iteration, we extract negative samples from the HTML documents in the test fold and use the synthetic negative samples under evaluation on the remaining folds. Subsequently, we employ the positive samples and synthetic negative samples from the training folds to train a classifier, which is then evaluated on the data of the test fold.

Table A1 shows the results obtained for all these possible combinations. These results confirm that the most effective combination includes *random match* (both monolingual and bilingual), *maximize Jaccard similarity* (both monolingual and bilingual), and *remove random token* (bilingual only), with a macro F1 of 95.74% (positive class F1: 92.00%; negative class F1: 99.49%).

Table A1. Results of the evaluation of all combinations of methods to generate synthetic negative samples. The best combination appears in boldface

Methods[a]	Pos. F1	Neg. F1	Macro F1	Methods[a]	Pos. F1	Neg. F1	Macro F1
1	75.31%	98.02%	86.66%	2 + 3 + 5	88.45%	99.24%	93.85%
2	85.49%	99.03%	92.26%	2 + 3 + 6	84.88%	98.99%	91.93%
3	27.87%	81.93%	54.90%	2 + 4 + 5	88.63%	99.25%	93.94%
4	79.73%	98.37%	89.05%	2 + 4 + 6	88.78%	99.27%	94.02%
5	58.43%	95.68%	77.06%	2 + 5 + 6	91.22%	99.44%	95.33%
6	20.50%	67.82%	44.16%	3 + 4 + 5	90.42%	99.35%	94.89%
1 + 2	84.07%	98.89%	91.48%	3 + 4 + 6	86.58%	99.06%	92.82%

https://doi.org/10.1017/nlp.2026.10021 Published online by Cambridge University Press

Table A1. Continued

Methods[a]	Pos. F1	Neg. F1	Macro F1	Methods[a]	Pos. F1	Neg. F1	Macro F1
1 + 3	77.33%	98.24%	87.79%	3 + 5 + 6	78.82%	98.38%	88.60%
1 + 4	84.89%	98.93%	91.91%	4 + 5 + 6	90.87%	99.40%	95.14%
1 + 5	88.47%	99.24%	93.85%	1 + 2 + 3 + 4	89.21%	99.29%	94.25%
1 + 6	74.49%	97.94%	86.22%	1 + 2 + 3 + 5	89.40%	99.32%	94.36%
2 + 3	87.60%	99.16%	93.38%	1 + 2 + 3 + 6	89.22%	99.31%	94.27%
2 + 4	88.56%	99.25%	93.91%	1 + 2 + 4 + 5	88.03%	99.21%	93.62%
2 + 5	88.48%	99.24%	93.86%	1 + 2 + 4 + 6	87.83%	99.18%	93.51%
2 + 6	88.00%	99.22%	93.61%	1 + 2 + 5 + 6	90.40%	99.39%	94.90%
3 + 4	88.96%	99.26%	94.11%	1 + 3 + 4 + 5	90.72%	99.39%	95.06%
3 + 5	82.39%	98.74%	90.56%	1 + 3 + 4 + 6	83.84%	98.86%	91.35%
3 + 6	24.87%	76.66%	50.77%	1 + 3 + 5 + 6	91.27%	99.44%	95.35%
4 + 5	62.00%	96.29%	79.14%	1 + 4 + 5 + 6	91.52%	99.45%	95.48%
4 + 6	69.23%	97.22%	83.22%	2 + 3 + 4 + 5	89.59%	99.32%	94.45%
5 + 6	76.93%	98.18%	87.55%	2 + 3 + 4 + 6	88.17%	99.21%	93.69%
1 + 2 + 3	88.98%	99.27%	94.12%	2 + 3 + 5 + 6	89.22%	99.30%	94.26%
1 + 2 + 4	85.39%	98.98%	92.18%	2 + 4 + 5 + 6	90.24%	99.36%	94.80%
1 + 2 + 5	88.41%	99.25%	93.83%	3 + 4 + 5 + 6	89.73%	99.31%	94.52%
1 + 2 + 6	86.76%	99.12%	92.94%	**1 + 2 + 3 + 4 + 5**	**92.00%**	**99.49%**	**95.74%**
1 + 3 + 4	84.53%	98.92%	91.72%	1 + 2 + 3 + 4 + 6	85.15%	99.01%	92.08%
1 + 3 + 5	90.86%	99.41%	95.14%	1 + 2 + 3 + 5 + 6	90.52%	99.39%	94.95%
1 + 3 + 6	70.89%	97.48%	84.18%	1 + 2 + 4 + 5 + 6	87.93%	99.19%	93.56%
1 + 4 + 5	90.14%	99.35%	94.74%	1 + 3 + 4 + 5 + 6	89.06%	99.29%	94.17%
1 + 4 + 6	84.48%	98.89%	91.69%	2 + 3 + 4 + 5 + 6	91.15%	99.43%	95.29%
1 + 5 + 6	89.58%	99.32%	94.45%	1 + 2 + 3 + 4 + 5 + 6	88.87%	99.27%	94.07%
2 + 3 + 4	90.38%	99.38%	94.88%				

[a] Codification: *random match bilingual* (1), *maximize Jaccard similarity bilingual* (2), *remove random tokens bilingual* (3), *random match monolingual* (4), *maximize Jaccard similarity monolingual* (5), *remove random tokens monolingual* (6).

Cite this article: García-Romero C, Esplà-Gomis M and Sánchez-Martínez F. Smart bilingual focused crawling of parallel documents. *Natural Language Processing* https://doi.org/10.1017/nlp.2026.10021