Harvesting parallel text in multiple languages with limited supervision

Luciano Barbosa[1] *Vivek Kumar Rangarajan Sridhar*[1]
Mahsa Yarmohammadi[2] *Srinivas Bangalore*[1]
(1) AT&T Labs - Research, 180 Park Avenue, Florham Park, New Jersey 07932
(2) Oregon Health & Science University, Beaverton, Oregon 97006
`lbarbosa,vkumar,srini@research.att.com, yarmoham@ohsu.edu`

Abstract

The Web is an ever increasing, dynamically changing, multilingual repository of text. There have been several approaches to harvest this repository for bootstrapping, supplementing and adapting data needed for training models in speech and language applications. In this paper, we present semi-supervised and unsupervised approaches to harvesting multilingual text that rely on a key observation of *link collocation*. We demonstrate the effectiveness of our approach in the context of statistical machine translation by harvesting parallel texts and training translation models in 20 different languages. Furthermore, by exploiting the *DOM trees* of parallel webpages, we extend our harvesting technique to create parallel data for resource limited languages in an unsupervised manner. We also present some interesting observations concerning the socio-economic factors that the multilingual Web reflects.

Keywords: web crawling, parallel text, document model object tree, machine translation.

1 Introduction

The amount of information and knowledge in the World Wide Web (WWW) is increasing at a rapid rate. As of September 2011, approximately 500 million websites were estimated to be present in the WWW; a jump of 1000% from 1995 [1]. An increasingly large proportion of these websites are in non-English languages. For example, the total proportion of English webpages on the WWW has been estimated to have dropped from 80% in 1996 to about 45% in 2008 (Pimienta et al., 2009). Furthermore, only 27% of the internet users claim English as their native language (Miniwatts Marketing Group, 2011). The opportunity to capitalize on this new market has encouraged internet service providers to provide web content in multiple languages. As a result, the Web has become an attractive resource for acquiring structured and unstructured data across several low-resource languages and domains.

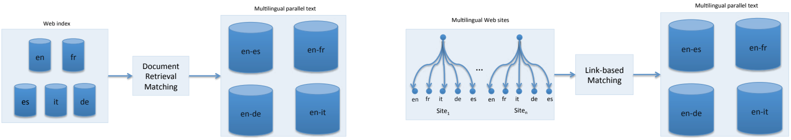
Multilingual web data has been especially useful in a variety of natural language processing tasks such as information retrieval, language modeling and machine translation. Statistical machine translation requires the creation of a large corpus of parallel text, i.e., translations of text across languages. Harvesting such data automatically by exploiting multilingual webpages provides a low-cost, scalable alternative to expensive expert human translations. Moreover, bilingual subject matter experts may be extremely difficult to find for certain

[1] http://news.netcraft.com/archives/2011/09/06/september-2011-web-server-survey.html

language pairs. As a result, web harvesting of parallel text has been addressed extensively in the recent past (Resnik and Smith, 2003; Shi et al., 2006; Pomikálek, 2008; Utiyama et al., 2009; Hong et al., 2010; Almeida and Simões, 2010; Uszkoreit et al., 2010).

Parallel text acquisition can be performed by examining the Web in either an unstructured or structured way. In the unstructured view of the Web, a large Web index is used as a starting point and the webpages are matched using cross-language document retrieval (see Figure 1(a)). The basic idea behind such an approach is to use a seed translation model to translate the entire index in one particular language and then match the pages using document retrieval techniques (Uszkoreit et al., 2010). The feasibility of such an approach is dependent on the availability of a large Web index as well as computational power to perform large scale machine translation and document alignment. In the structured approach, websites containing parallel text are typically identified using search engine APIs and the crawler proceeds to identify new websites based on the preceding ones (Resnik and Smith, 2003; Shi et al., 2006; Utiyama et al., 2009; Hong et al., 2010). However, such a scheme is amenable primarily for pair of languages.

Our work takes a structured view of the Web and exploits the link structure of websites to collect multilingual parallel text. We leverage the property that multilingual websites typically provide content simultaneously in several languages. Furthermore, a link that represents an entry point to a particular language in these websites usually co-occurs in the DOM tree with entry points to other languages (see Figure 1(b)). The co-occurring language versions of webpages on a particular website is influenced by geographic and economic factors of the underlying service or business. For example, a hospitality website identified as a possible source for harvesting English-French parallel text may also contain German, Italian and Spanish versions of the site, whereas a website with English-Chinese parallel text may have corresponding Japanese and Korean counterparts.



(a) Unstructured view of Web: Web index based parallel text acqusition

(b) Structured view of Web: Multilingual crawling using language co-occurrence

Figure 1: Strategies for harvesting parallel text across multiple languages from the Web

In this work, we present a framework for harvesting parallel text across multiple language pairs using the approach illustrated in Figure 1(b). We present semi-supervised and unsupervised approaches to harvesting multilingual text that rely on a key observation of *link collocation*, i.e., entry points to different languages on multilingual websites are often collocated on the HTML DOM tree. Subsequently, we use an intra-site crawler and a suite of alignment procedures to generate parallel text across multiple pairs of languages and evaluate their utility in machine translation. We demonstrate significant improvements in translation quality for almost all of the 20 language pairs (with English as the source language) in the Europarl corpus (Koehn, 2005) through the addition of parallel text harvested through our approach. We also report experiments in English-Hindi translation by using a completely unsupervised approach.

The rest of the paper is organized as follows. In Section 2, we describe related work in web mining for parallel text and contrast our work with prior efforts. We describe the semi-supervised approach for obtaining multilingual entry points in a website in Sections 3 and 4 followed by a description of the overall framework for harvesting parallel text from the entry points. In Section 6 we present statistical machine translation experiments using the data harvested through our framework and present a detailed case study for English-Hindi translation in Section 7. We provide a brief discussion in Section 8 and conclude in Section 9 along with directions for future work.

2 Related Work

Prior research on acquiring parallel text from the Web has also focused primarily on a specific pair of languages. Even though in principle the algorithm and framework can be extended to other pairs of languages, it requires either parallel webpages or comparable documents to trigger the process. For example, (Resnik and Smith, 2003) used a crawling procedure to harvest parallel text in English-Arabic starting from the Internet Archive and using several language specific patterns in the URLs. The work in (Munteanu and Marcu, 2005) matches comparable documents in English-Chinese and English-Arabic and subsequently extracts parallel text while (Fung and Cheung, 2004) extract parallel text from quasi-comparable documents in English-Chinese. The work in (Hong et al., 2010; Shi et al., 2006) uses several web crawling strategies for harvesting parallel text in English-Chinese and the work in (Utiyama et al., 2009) addresses the extraction of Japanese-English parallel text from mixed language pages.

Conventionally, the crawling procedure to detect websites containing parallel text has been through a query-based approach (Resnik and Smith, 2003; Chen and Nie, 2000; Hong et al., 2010). They submit carefully constructed queries to a search engine that might yield potential parallel webpages. However, such a procedure depends on the quality of the query strategy as well as the Web index provided by the search engine. Furthermore, the process is typically constrained for particular pair of languages. Subsequently, the websites are mined independently and matched through a variety of techniques ranging from document retrieval (Munteanu and Marcu, 2005) to matching based on HTML document object model (DOM) tree similarity (Resnik and Smith, 2003; Shi et al., 2006). Once the documents are aligned, the sentences are aligned next using dynamic programming based on sentence length and a bilingual dictionary (Resnik and Smith, 2003) or through a classifier (Munteanu and Marcu, 2005).

Multilingual parallel text extraction has been specifically addressed in (Uszkoreit et al., 2010). However, the starting point for harvesting parallel text is a large Web index that is aligned using a seed translation model and subsequent document matching. It may not be feasible to acquire such a large index or run computationally expensive document matching for many efforts. Furthermore, their work reports translation quality experiments for about 7 language pairs and requires a seed translation model from the language of interest into English. Our interest, on the other hand, lies in crawling the Web to detect multilingual pairs of entry points belonging to websites that contain parallel text.. We push the task of identifying parallel web pages upfront to the crawler instead of downstream document matching over a large snapshot of the Web. Our framework can be used with or without a seed bilingual dictionary or translation model. In the absence of a seed dictionary, we use an unsupervised DOM tree similarity procedure to harvest parallel text as we describe later.

3 Background: Bilingual crawler

The simplest manifestation of a multilingual crawler is for a pair of languages, i.e., bilingual crawling. To perform this task, the crawler needs to detect bilingual sites by traversing interesting regions of the Web. The bilingual site detector (BiSite detector) is the component responsible for determining whether a website contains bilingual content. The detector performs its task in two phases: link-based prediction and language identification. The role of the *link predictor* is to predict links that are entry points to a particular language in a website. The *link predictor* relies on the property that these entry points contain some common link pattern. For instance, entry points to the French content might have words as "fr" or "francais" in their URLS. In order to be able to handle different types of patterns in the links, it uses features in 5 different contexts: tokens in the URL, anchor, around the link, image alt and image src tags. Thus, for each language, a link predictor is built using supervised learning. Subsequently, the BiSite detector verifies if the pages whose links were considered relevant by the link predictor are in the languages of interest. Once a pair of links in a website are hypothesized as entry points in two different languages, the crawler uses an intra-site crawling policy similar to that described in (Rangarajan Sridhar et al., 2011) to traverse the Web sites and collect the parallel content. Figure 2 depicts a simple illustration of the bilingual crawler. Further details about the BiSite detector is presented in (Barbosa et al., 2011).

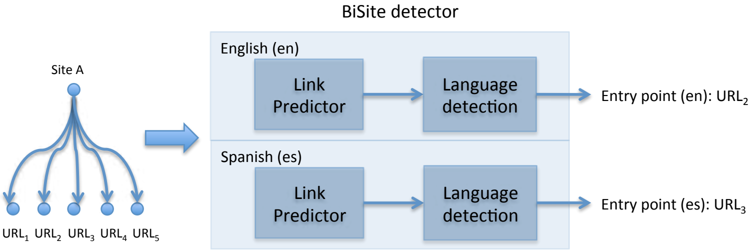


Figure 2: BiSite detector detects entry points to parallel text for a pair of languages on a given website.

4 From bilingual to multilingual crawling

The task of a multilingual crawler is to detect websites that contain content in multiple languages. A simple way to perform this task is to extend the approach used in the BiSite detector by building entry point detectors for each language of interest. The BiSite detector was built by labeling positive and negative examples of entry points in a pair of languages. Although effective, a fully supervised approach is not feasible for building entry point detectors for many pairs of languages as it requires significant labeling effort.

We propose a semi-supervised approach for building multilingual detectors. Our approach relies on the observation that, in a given website, entry points to different languages are collocated links on the same page. The bootstrapping algorithm works as follows: BiSite detectors in a small set of language pairs are constructed using manual labeling and are in turn used to identify entry points in these languages. The algorithm then extracts links collocated with the detected entry points, generating training data to build detectors in new language pairs. The new detectors can now be used in the first phase of the bootstrapping, and iterated to generate more entry points. Figure 3 describes the components of the bootstrapping algorithm. Therefore, the only supervision provided is positive (entry points) and negative examples for the initial detectors whose accuracy the bootstrapping algorithm heavily relies on.

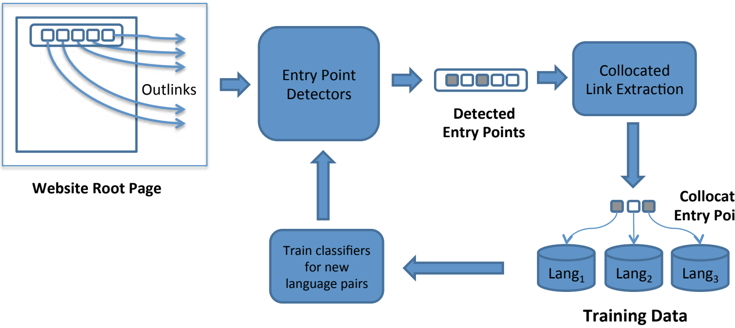


Figure 3: Bootstrapping algorithm for creating classifiers for new pairs of languages

4.1 Extraction of co-occurring links

The extraction of co-occurring links is an important step in the bootstrapping algorithm. Its goal is to extract candidate entry points collocated with entry points identified by the initial BiSite detectors. Formally, let OL be the set of *outlinks*[2] on a given Web page WP, and EP be the set of language entry points in WP ($EP \subseteq OL$). The algorithm's objective is to identify EP, if it exists. Our assumption is that this subset is the one with highest diversity with respect to languages among OL. In other words, while most of the outlinks in OL point to pages in similar languages, the links in EP point to pages in distinct languages. The algorithm then searches for the partition of OL, $\widehat{EP}$, with the maximum normalized entropy with respect to its languages among the subsets ($\mathbf{SL} = \{SL_1, \cdots, SL_K\}$) of OL:

$$\widehat{EP} = \operatorname*{arg\,max}_{\mathbf{SL_i} \in \mathbf{SL}} \; NormEntropy(\mathbf{SL_i}) \tag{1}$$

$$NormEntropy(SL_i) = \frac{-\sum_{k=1}^{|L_i|} p(l_k) log(p(l_k))}{log(|SL_i|)} \tag{2}$$

where $L_i = l_{i1}, ... l_{i|L|}$ is the set of the languages in SL_i and $p(l_k) = count(l_k)/|SL_i|$. NormEntropy is equal to 1 when SL_i is composed of distinct languages.

Since finding an optimal subset of links is a combinatorial search problem, our algorithm adopts a greedy approach. The algorithm imposes the following constraints to make the

[2] *Outlinks* are the links pointing out from a given webpage.

solution tractable: (1) $\widehat{EP}$ must contain the entry points, $iniEPs$, detected by initial detectors; (2) the elements of $\widehat{EP}$ are not spread over the entire Web page but located in a similar region in the page's DOM tree. Starting from a detected entry point $iniEP_i$, the algorithm first locates the DOM node ($node_i$), represented by the "a href" HTML tag, associated to $iniEP_i$ and calculates the normalized entropy of the subset of OL contained in the DOM subtree of the parent of $node_i$. In the next step, it goes up one level in the tree and calculates the normalized entropy of the subtree. If the normalized entropy of the current subtree is higher than the previous one, it continues the search up one more level, otherwise stops. The links of the best subtree identified by this process SL_i are considered as the candidates associated with $iniEP_i$. This procedure is repeated for each $iniEP_i$ and the set SL_i with the highest entropy among the candidate sets is considered for the final step. Subsequently, the algorithm discards elements in SL_i that are from the same language as we assume that there is only one single entry point for each language in a given page. Finally, the output of this process is a set of candidate entry points (outlinks) in which each outlink is associated to a particular language. For each outlink, its link neighborhood is extracted (described previously) and used as a positive example for building a *link predictor* for that particular language.

4.2 Evaluation of semi-supervised multilingual crawler

In this section, we assess the quality of the entry points identified through our bootstrapping algorithm. As inputs to the algorithm we provided two entry point detectors (English and Spanish), created using labelled data, and 10,000 Web sites, collected using an unrestricted crawler. We then ran the algorithm over these Web sites and candidate entry points were extracted in 45 different languages. Figure 4 shows the top-10 languages collocated with English and Spanish in this dataset. The most popular languages were European languages while Japanese was the most popular Asian language, beating European languages such as Polish and Slovenian.

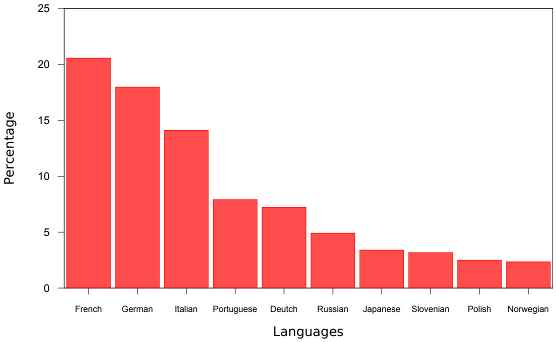


Figure 4: Distribution of languages collocated with English and Spanish

For 5 of the top-10 languages: French, German, Portuguese, Russian and Japanese, we manually labelled test data (about 500 positive and 1,000 negative examples for each language) to evaluate the performance of their respective detectors created using the semi-supervised approach. We also assessed the accuracy of the co-occurring link extraction (CLE) by manually inspecting how many entry points extracted from the CLE procedure were correct. Table 1 presents the precision of CLE and F-measure of the detectors. For each of the top 5 languages the collocated link extraction algorithm achieves greater than 90% precision. The detectors created automatically by the bootstrapping algorithm have a high F-measure (from 0.8 to 0.87). This is a direct consequence of the high precision obtained by the CLE algorithm that provides the positive examples for training the classifiers.

Language	**F-measure (Detector)**	**Accuracy (CLE)**
German	0.87	0.96
French	0.85	0.93
Japanese	0.85	0.95
Russian	0.8	0.98
Portuguese	0.8	0.95

Table 1: Precision and F-measures for the detectors constructed using the semi-supervised entry point detector approach

5 Parallel text acquisition

The multilingual crawler generates pairs of entry points for multiple language pairs that subsequently need to be mined for parallel text. We adopt a recursive intra-site crawling approach that aligns text and URLs across the initial entry points to harvest parallel text. Our framework uses a document matching framework to align the URLs (Munteanu and Marcu, 2005) and a bilingual dictionary based dynamic programming match to align the sentences across the hypothesized parallel documents (Ma, 2006). The document matching process is constrained by a window size that is dependent on the total number of parallel URLs that need to be aligned. The framework also enables us to highly parallelize mining across multiple language pairs. The bilingual dictionary in this work was obtained by performing automatic word alignment on seed parallel data (Och and Ney, 2003). The harvested bitext was then filtered using a word-overlap filter as well as a source and target vocabulary restriction filter. By varying the thresholds for the various filters, one can control the amount and quality of the bitext. We also check for the fidelity of the translation by matching a subset of the harvested text for each website against translations from Google Translate and Microsoft Bing. We omit the data from the entire website if the translations have a high correlation with the online translation engines (cosine distance ≥ 0.8). This step is performed to avoid the use of machine translated parallel text. A detailed description of the intra-site mining procedure can be found in (Rangarajan Sridhar et al., 2011).

6 Machine translation experiments

In this section, we validate the quality of the parallel text obtained using our multilingual crawling approach through machine translation experiments. Our objective is to evaluate the translation quality with and without the parallel text harvested from the Web for a large number of language pairs. We used the Moses[3] toolkit (Koehn et al., 2007) for performing phrase-based translation experiments. The standard pipeline (sentence alignment using GIZA++, phrase extraction with maximum phrase length of 7 using *grow-diag-final* option, lexicalized reordering model with *msd-bidirectional-fe* option) was used to build the

[3] http://www.statmt.org/moses

models. The language models were interpolated Kneser-Ney discounted trigram models, all constructed using the SRILM toolkit (Stolcke, 2002). The language models were constructed only from the target side of the bitext for each language pair. We performed minimum error rate training (MERT) on a development set to optimize the weights of the log-linear model.

We ran machine translation experiments for all of the 20 languages (English as source language) present in the Europarl corpus (Koehn, 2005). The baseline models were trained on Europarl data. For each of the 20 languages, we harvested parallel text (see Section 5) from the entry points hypothesized by the multilingual crawler. Statistics of the parallel text obtained using our procedure is shown in Table 2. Subsequently, we trained a translation model by combining the Europarl text with the parallel text harvested using our approach.

Language pairs	# websites	# webpages	# parallel sentences	Language pairs	# websites	# webpages	# parallel sentences
en-bg	253	290	825	en-it	14538	79582	1654730
en-cs	5144	13058	250598	en-lt	200	1154	30725
en-da	3384	9678	138482	en-lv	128	298	10771
en-de	33582	117043	1215186	en-nl	8295	35109	568534
en-el	705	1908	9645	en-pl	1894	7184	154508
en-es	13075	56687	1347795	en-pt	3542	11900	268065
en-et	288	1766	37141	en-ro	767	2627	51498
en-fi	682	1680	20355	en-sk	443	2556	87172
en-fr	15429	54315	1140140	en-sl	327	2001	59935
en-hu	1481	6492	129291	en-sv	3390	14382	247326

Table 2: Parallel text obtained using our framework for the 20 languages in Europarl corpus

Figure 5 shows the BLEU scores across the 20 languages with and without the addition of parallel text from the Web. We performed the decoding experiments on three different test sets: Europarl, Web and Europarl+Web. We chose the three sets to represent a variety of test domains. A total of 1000 sentences was used for testing. The test sentences from the Web were chosen randomly such that they satisfied a sentence length constraint (≥ 5 words). We spot checked a small subset of the test sets to ascertain the fidelity. Ideally, a manually constructed general vocabulary test set for each of the language pairs would have been the best choice. However, it is expensive and difficult to create test sets through human experts for each of the 20 language pairs. All the models are optimized based on a randomly chosen development set from Europarl comprising of 1000 sentences.

Overall, the results in Figure 5 indicate significant improvements in BLEU scores when the harvested parallel text is added to the Europarl data. As expected, the improvements are not quite marked when tested on Europarl data, however, they are significant on data obtained from a general domain (a mix of websites). Our objective is to translate sentences closer to a general domain (hospitality, business, medicine, etc.) and we show that parallel data harvested from the Web provides significant improvement.

7 Unsupervised parallel text acquisition: English-Hindi

In the previous section, we demonstrated the utility of our multilingual crawler for languages with reasonable resources. However, such data may not be present for several other language pairs. One of the main objectives of our Web crawling scheme is to harvest parallel text

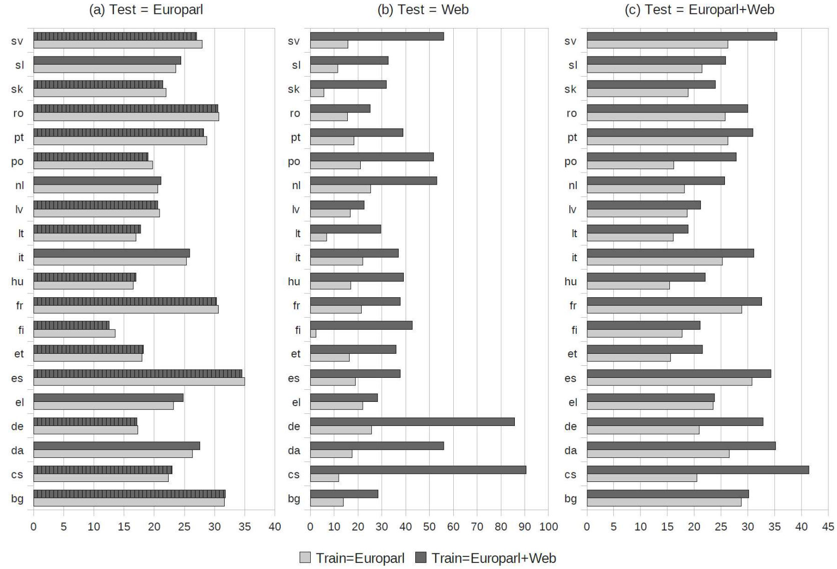
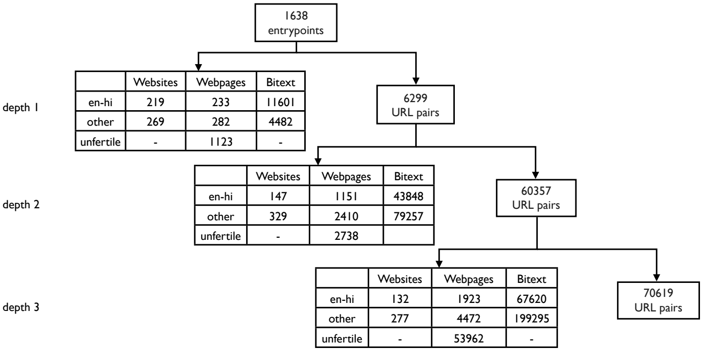


Figure 5: Translation quality as measured through BLEU score for various test sets with and without web crawled parallel text. Hatched bars indicate insignificant difference (Koehn, 2004) with respect to the baseline model built from Europarl data.

for language pairs with limited resources, i.e., lack of publicly available large database or language resources. As an instantiation of this goal, we conducted a detailed study on English-Hindi. We used the collocated link extraction procedure described in Section 4.1 to compile 1638 potential entry points in English-Hindi. Unlike the language pairs used in the previous section, we did not have access to parallel text or a bilingual dictionary for English-Hindi. Hence, we used a completely unsupervised scheme to harvest parallel text from the initial entry points. The intra-site crawler (see Section 5) was modified to perform document matching using the HTML structure of the webpages and the dynamic programming text alignment procedure relied only on sentence length and identity anchor words (words that are present in the source and target sentence). We computed the distance between the DOM trees of two HTML pages to decide if the pages contained parallel text (Pawlik and Augsten, 2011). We ran the crawler for three iterations, each iteration using the parallel entry points identified in the previous step. Figure 6 shows the number of Web sites, pages and bitext harvested using the setup. Since, Hindi characters have a pre-defined range, we could filter out bitext that did not contain Hindi.

From Figure 6 it can be seen that the growth of the entry points hypothesized at each iteration of the intra-site crawling procedure is not linearly related to the bitext harvested. We conjecture that websites containing English-Hindi parallel text do not point to significant number of newer English-Hindi websites. Unlike European languages the domain of the WWW containing English-Hindi parallel text is rather limited.



depth 1

	Websites	Webpages	Bitext
en-hi	219	233	11601
other	269	282	4482
unfertile	-	1123	-

depth 2

	Websites	Webpages	Bitext
en-hi	147	1151	43848
other	329	2410	79257
unfertile	-	2738	

depth 3

	Websites	Webpages	Bitext
en-hi	132	1923	67620
other	277	4472	199295
unfertile	-	53962	-

Figure 6: Illustration of the number of websites, webpages and bitext harvested in an unsupervised manner for English-Hindi. *other* refers to bitext in a language pair other than English-Hindi and *unfertile* refers to entry points that did not harvest any bitext.

We also performed machine translation experiments using the English-Hindi parallel text harvested through the unsupervised alignment approach. The baseline model was trained on the Indic multi-parallel corpus (Birch et al., 2011) and a new model was trained by adding the harvested parallel text. Since the parallel text harvested using the unsupervised approach is prone to be noisy, we filtered the sentences using a word-overlap filter constructed from the IBM Model1 dictionary obtained from the baseline translation model. We used the Indic corpus development and test sets for tuning and testing, respectively. The results are reported in Table 3. The results demonstrate a significant improvement ($p = 0.05$) in BLEU score when the Web crawled parallel text is added to the baseline data. The filtering procedure using the dictionary obtained through automatic alignment yields lesser amount of parallel text and hence results in smaller improvement in BLEU score. The experiments clearly indicate the benefit of the parallel text harvested using our scheme. It is important to note that we started this process for English-Hindi with no resources whatsoever.

Training data	**BLEU**
Indic training data	18.6
Indic+unsupervised parallel text	19.4
Indic+ parallel text filtered using IBM Model1 dictionary	19.1

Table 3: BLEU scores on English-Hindi corpus using parallel text harvested form multilingual crawler.

8 Discussion

Languages addressed by multilingual crawler. The multilingual crawler presented in this work generates entry points for over 500 language pairs (with atleast 500 entry points). We have presented machine translation experiments only for a subset of these languages, namely, ones contained in the Europarl corpus and English-Hindi. Some of the prominent language pairs with a large number of entry points are English-Chinese (26893), English-Japanese (25804), English-Russian (24011), English-Turkish (10879), English-Norwegian (6564), English-Arabic (5001), English-Korean (4425), English-Persian (3398), English-

Indonesian (1224), English-Hebrew (1514) and English-Thai (1000). One can potentially harvest parallel text for all of these language pairs and subsequently exploit it for machine translation. Other notable language pairs not containing English as one of the languages are German-French (18874), German-Italian (16275), German-Spanish (14410), German-Dutch (10530), and French-Spanish (16295). It is interesting to note that we can obtain entry points for a multitude of language pairs not containing English. Directly obtaining parallel text for some of these language pairs can obviate the need to use English as a pivot language during translation.

Distribution of parallel text on the Web in European languages. Based on the parallel text extracted for the language pairs in the Europarl data (see Table 2), English-Bulgarian (en-bg) is the poorest language pair in terms of net harvested parallel text. The total bitext of 825 sentences is obtained from 253 websites, i.e., the yield per website is only about 3 sentences. Most of the websites that contained any Bulgarian translations did so only at the top level (menu items, homepage information, etc.) and did not have any significant content translated across internal webpages. English-Greek (en-el) is another language pair with very low parallel content on the Web. English-Estonian (en-et), English-Lithuanian (en-lt), English-Slovakian (en-sk) and English-Slovenian (en-sv) are the language pairs with the highest density of parallel text per website. While there are not many websites with parallel text in these languages, the ones that do contain them are very fertile. As expected English-French (en-fr), English-Spanish (en-es), English-German (en-de), and English-Italian (en-it) language pairs produce large amounts of parallel text in comparison with other languages due to the dominance of these European languages on the Web. Our experiments also indicate that on an average there are approximately 3-6 unique webpages for each website that yields parallel text.

Language co-occurrence. Figure 7 presents a graph showing how the languages are collocated in our data. The edges between languages represent that entry points of these languages are collocated on the same websites. Given a language, we calculated the proportion of languages collocated with it. The edge weight in the graph represents this proportion. Since almost all languages are collocated with each other, we pruned some edges to make the figure clearer. For that, we removed edges with weights lower than 0.05, deleted the English node, since it is the most popular collocated language, and some languages from middle Europe. First thing one can note is that German is the most popular language after English. One can also observe there are some language clusters mainly due to geographic or social/economic aspects: the Western European (German, French, Spanish, Italian, Dutch and Portuguese), the Eastern European (Russian, Estonian, Ukrainian, Latvian, Lithuanian and Bulgarian), the Middle Eastern (Arabic, Urdu and Farsi) and the Far East Asian (Japanese, Korean, Chinese Mandarin and Taiwanese Mandarin). Japanese has a high collocation with European languages because many European websites provide content in Japanese. There is a high collocation of German with Turkish. We believe the reason for that is the high population of Turkish in Germany.

Socio-linguistic observations. Another study that we performed was to obtain the distribution of the languages in the multilingual websites per each country (here we relied on country top-level domain codes). There are countries where very few multilingual sites are available, e.g., India and Cuba, whereas others such as Germany, Netherlands, China and Japan provide a great number of multilingual sites. As one expected, English is the

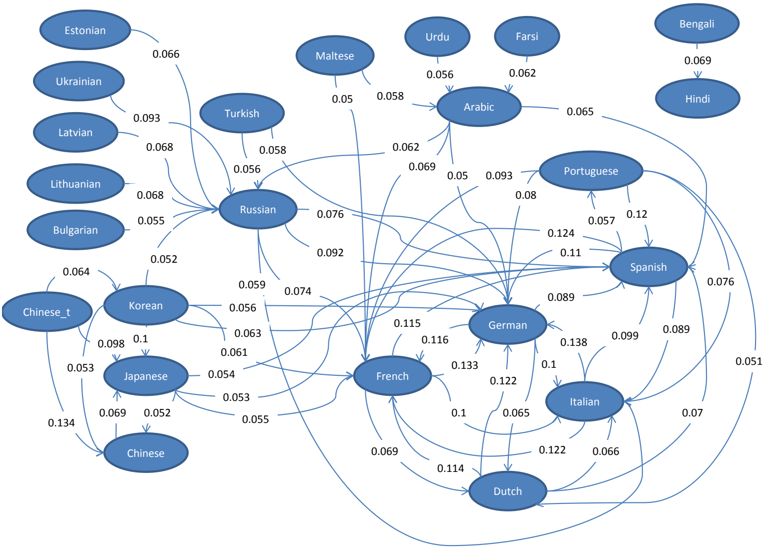


Figure 7: Distribution of language collocations.

foreign language most popular in all countries. It is interesting to note that the popularity of collocated languages in a country depends on geographic and social/economic aspects. Here are some examples:

- Geographic: the most popular foreign languages in Czech multilingual websites (.cz) are in this order: English, German, Russian, Slovak and Polish. These languages are either from neighbouring countries or from economic powers. Interestingly, however Czech is not a very popular language in Polish websites (.pl).
- Social/economic: In Israel, the number of websites in Russian is comparable to Hebrew and English, due to the Russian immigrants. In Iran, French is the most popular language after English, Farsi and Arabic, for historical reasons. Portuguese is the most popular European language in Japanese websites (probably to reach the huge number of Brazilian immigrants that live there).

9 Conclusion

We presented a novel semi-supervised approach for detecting parallel text across multilingual websites. Our approach takes a structured view of the Web and crawls regions of the Web that are likely to produce significant amount of parallel text. First, we constructed supervised classifiers for a few language pairs to detect websites with potential parallel text. By exploiting the property that in many multilingual websites, entry points to different languages are often collocated on the HTML DOM tree, we use a collected link extraction algorithm to extract entry points for new language pairs. Subsequently, the data is used to train supervised classifiers to detect entry points for new language pairs. Starting from a classifier trained to detect English-Spanish entry points, we were able to obtain entry points in 45 language pairs. We used an intra-site crawling approach to mine the entry points and align the text using document retrieval and dynamic programming techniques. We demonstrated significant improvements in translation quality for all of the 20 languages in the Europarl corpus. We contrasted the experiments conducted on Europarl with those performed on English-Hindi that did not have any resources to learn a seed translation model or dictionary for sentence alignment. Finally, we also presented some socio-linguistic observations inferred through our crawling procedure. We plan to conduct experiments on the other 24 language pairs not reported in this work as part of future work. We are interested in acquiring parallel text for languages with no or low resources and identifying websites that can be mined frequently due to their dynamic nature (e.g., news, broadcasting stations, etc.).

References

Almeida, J. and Simões, A. (2010). Automatic parallel corpora and bilingual terminology extraction from parallel web sites. In *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora*.

Barbosa, L., Bangalore, S., and Rangarajan Sridhar, V. K. (2011). Crawling back and forth: Using back and out links to locate bilingual sites. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 429–437, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Birch, L., Callison-Burch, C., Osborne, M., and Post, M. (2011). The indic multi-parallel corpus. `http://homepages.inf.ed.ac.uk/miles/babel.html`.

Chen, J. and Nie, J. (2000). Parallel web text mining for cross-language IR. In *RIAO*, volume 1, pages 62–78.

Fung, P. and Cheung, P. (2004). Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of the 20th international conference on Computational Linguistics*.

Hong, G., Li, C.-H., Zhou, M., and Rim, H.-C. (2010). An empirical study on web mining of parallel data. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., W., S., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*.

Ma, X. (2006). Champollion: A robust parallel text sentence aligner. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*, Genova, Italy.

Miniwatts Marketing Group (2011). Number of internet users by language. *Internet World Stats.*

Munteanu, D. S. and Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Comput. Linguistics*, 29(1):19–51.

Pawlik, M. and Augsten, N. (2011). RTED: A robust algorithm for the tree edit distance. *Proceedings of VLDB Endowment*, 5(4):334–345.

Pimienta, D., Prado, D., and Blanco, A. (2009). Twelve years of measuring linguistic diversity in the internet: balance and perspectives. *Paris: Unesco.*

Pomikálek, J. (2008). *Building parallel corpora from the Web.* PhD thesis, Masarykova univerzita.

Rangarajan Sridhar, V. K., Barbosa, L., and Bangalore, S. (2011). A scalable approach to building a parallel corpus from the Web. In *Proceedings of Interspeech.*

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

Shi, L., Niu, C., Zhou, M., and Gao, J. (2006). A DOM tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.*

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP.*

Uszkoreit, J., Ponte, J. M., Popat, A. C., and Dubiner, M. (2010). Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics.*

Utiyama, M., Kawahara, D., Yasuda, K., and Sumita, E. (2009). Mining parallel texts from mixed-language web pages. In *Proceedings of MT Summit.*