# Crawling Back and Forth: Using Back and Out Links to Locate Bilingual Sites

**Luciano Barbosa**
AT&T Labs – Research
180 Park Ave
Florham Park, NJ 07932
lbarbosa@research.att.com

**Srinivas Bangalore**
AT&T Labs – Research
180 Park Ave
Florham Park, NJ 07932
srini@research.att.com

**Vivek Kumar Sridhar Rangarajan**
AT&T Labs – Research
180 Park Ave
Florham Park, NJ 07932
vkumar@research.att.com

## Abstract

This paper presents a novel crawling strategy to locate bilingual sites. It does so by focusing on the Web graph neighborhood of these sites and exploring the patterns of the links in this region to guide its visitation policy. A sub-task in the problem of bilingual site discovery is the job of detecting bilingual sites, i.e., given a Web site, verify whether it is bilingual or not. We perform this task by combining supervised learning and language identification. Experimental results demonstrate that our crawler outperforms previous crawling approaches and produces a high-quality collection of bilingual sites, which we evaluate in the context of machine translation in the tourism and hospitality domain. The parallel text obtained using our novel crawling strategy results in a relative improvement of 22% in BLEU score (English-to-Spanish) over an out-of-domain seed translation model trained on the European parliamentary proceedings.

## 1 Introduction

Parallel texts are translations of the same text in different languages. Parallel text acquisition from the Web has received increased attention in the recent years, especially for machine translation (Melamed, 2001) and cross-language information retrieval (Grossman and Frieder, 2004). For many years, the European Parliament proceedings (Koehn, 2005a) and official documents of countries with multiple languages were the only widely available parallel texts. Although these are high-quality corpora, they have some limitations: (1) they tend to be domain specific (e.g., government related texts); (2) they are available in only a few languages; and (3) sometimes they are not free or there is some restriction for using them. On the other hand, Web data is free and comprises data from different languages and domains.

Previous research in the area of parallel Web data acquisition has mainly focused on the problems of document pair identification (Jiang et al., 2009; Uszkoreit et al., 2010; Munteanu and Marcu, 2005; Resnik and Smith, 2003; Melamed, 2001) and sentence alignment. Typically, document pairs are located by issuing queries to a search engine (Resnik and Smith, 2003; Hong et al., 2010). The sentences in the matched documents are then aligned using standard dynamic programming techniques. In this work, we model the problem of obtaining parallel text in two subtasks. First, locate the sites that contain bilingual data (bilingual sites). Here we assume that parallel texts are present in the same site (Chen and Nie, 2000). Second, extract parallel texts within these sites. While the latter problem of extracting of parallel text from bilingual Web sites has received a lot of attention, the former problem of automatically locating high quality parallel Web pages is still an open problem.

In this paper, we propose a crawling strategy (Olston and Najork, 2010) to discover bilingual sites on the Web. Previous work on focused crawlers (Chakrabarti et al., 1999; Diligenti et al., 2000) has been used to locate different kinds of Web sources such as Web pages in a topic (Chakrabarti et al., 2002), geographic information (Ahlers and Boll, 2009) and Web forms (Barbosa and Freire, 2007) by following outlinks. In contrast to these approaches, we explore the idea of using not only forward links but also backlinks. *Backlinks* of a page $p$ are the links that point to $p$ and *outlinks (forward links)* are the links that $p$ points to. The reason for that is a single backlink page sometimes refers to many related pages, a phenomenon known as co-citation. Kumar et al. (Kumar et al., 1999) showed that co-



*Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 429–437,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

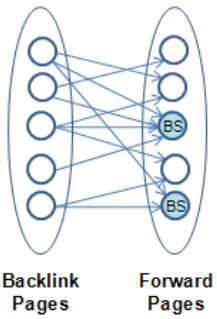


Figure 1: Bipartite graph representing the graph neighborhood visited by the crawler. Backlink pages point to pages in bilingual sites (BS) and other pages (forward pages).

citation is a common feature of Web communities and, as a result of that, Web communities are characterized by directed bipartite subgraphs. Based on that, we implemented our crawling strategy by restricting the crawler's search for bilingual sites in the bipartite graph composed by backlink pages (BPs) of the bilingual sites that were already discovered by the crawler, and pages pointed by BPs. This scheme is illustrated in Figure 1. Our assumption, therefore, is that the Web region represented by this bipartite graph is rich in bilingual sites since backlink pages typically point to multiple bilingual sites (co-citation). Finally, to focus on the most promising regions in this graph, the crawler explores the patterns in the links to guide its visitation policy.

A sub-task in the problem of bilingual site discovery is the job of detecting bilingual sites, i.e., given a Web site, verify whether it is bilingual or not. A simple approach to this task is to search the entire Web site for parallel text. However, this is computationally expensive since Web sites might contain hundreds/thousands of pages. We propose a low-cost strategy that visits very few pages in the Web site to make its prediction regarding the presence of bilingual text. Given a Web site's page, we use supervised learning to identify links on the page that are good candidates to point to parallel text within the site. Subsequently, our strategy verifies whether the pages pointed by the candidate links are in fact in the languages of interest.

The main contributions of this paper can be summarized as follows:

- A new focused crawling strategy that explores the concept of co-citation by restricting the search for targeted sources (bilingual sites in this paper) in the bipartite graph composed by the backlink pages of the targeted sources already discovered by the crawler, and the forward links pointed to by the backlink pages. The crawler uses link classifiers specialized in each set of the URLs of the pages (backward and forward pages) of the bipartite graph to focus on the most promising regions in this graph;
- A high-precision and efficient approach to detecting bilingual sites based on supervised learning and language identification.

The remainder of the paper is organized as follows. In Section 2, we present our approach to locating and detecting a bilingual site. We present experimental results in Section 3 and demonstrate the efficacy of our approach in the context of machine translation in Section 4. We review related work in Section 5 and conclude in Section 6.

## 2 Bilingual Site Crawler

A naive approach to collect parallel text would be to check for every pair of pages on the Web. However, this is computationally prohibitive given the scale of the Web. To make the search for parallel text more feasible, previous approaches made the assumption that parallel texts mainly occur within Web sites (Chen and Nie, 2000). Thus, the search for parallel text can be comprised of two steps. First, locate bilingual sites, and then, extract the parallel text from them. While previous approaches (Resnik and Smith, 2003; Zhang et al., 2006) have mainly focused on the latter problem of extracting sentence aligned parallel text from web, we are interested in the former problem of locating such sites.

The architecture of our crawler is presented in Figure 2. The crawler downloads a page, $p$ and sends it to the bilingual site detector (BS Detector). If the BS Detector predicts that the site represented by $p$ contains parallel text (see Section 2.1), the Backlink Crawler collects the backlinks of $p$, i.e., links that point to $p$, by using a search engine backlink API. The Backlink Classifier predicts the relevance of these links (see Section 2.3), and adds them to the queue that represent these links in the Frontier (backlink queue). The most promising backlink is then sent by the Frontier Scheduler to the Crawler, which downloads its content. Next, the Page Parser extracts the forward links of the backlink page and adds the most promising forward links (as identified by Forward-Link

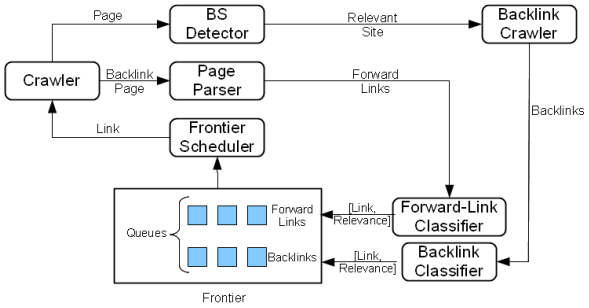


Figure 2: Architecture of our crawling strategy to locate bilingual sites.

Classifier) to the forward-link queue. The Frontier Scheduler then decides the next link to be sent to the crawler. We present the core elements of the crawler in the sections below.

### 2.1 Bilingual Site Detection

The performance of the bilingual site detection is essential to obtain a high-quality collection of bilingual sites. Zhang et al. (Zhang et al., 2006) perform this task by extracting the anchor text and image alt text from pages in the Web sites and match them with a pre-defined list of strings in the languages of interest. If the Web site contains at least two matched links in the different languages then it is considered as bilingual. The approach suffers from the drawback of low recall since bilingual sites that contain patterns outside the list may be missed. Another approach (Ma and Liberman, 1999) verifies the presence of bilingual text at pages of the top 3 or 4 levels of the Web site by using a language identifier. This approach can be very expensive as one might need to download a considerable portion of the Web site to make a decision.

Our solution to detecting parallel sites has some similarities with these previous approaches but tries to address their main limitations. First, instead of using a pre-defined list of patterns to detect bilingual sites, we use supervised learning to predict if a given page has links to parallel data (Link Predictor). Second, to avoid downloading a great portion of the Web site, the BS Detector only verifies whether the pages whose URLs are considered relevant by the Link Predictor are in different languages.

**Link-Based Prediction.** The role of the Link Predictor is to identify links that point to parallel text in a Web site. Our assumption is that pages of bilingual sites contain some common link patterns. For instance, pages in English might have a link to its version in Spanish, containing words such as "espanol" and "castellano" in its anchor, URL, etc. However, there are cases whereby the link does not provide any visible textual information to the user. Instead, only an image (usually a country flag) might represent the link. In these cases, textual information in the fields of the img html tag (e.g. alt and src) might be helpful. In order to be able to handle different types of patterns in the links, the Link Predictor uses features in 5 different contexts: tokens in the URL, anchor, around, image alt and image src. For this paper, we built the training data from non-bilingual and bilingual sites in English/Spanish. It was compiled by manually labeling 560 URLs (236 relevant and 324 non-relevant). We use probabilistic SVM (Platt, 1999) as the learning algorithm to create the Link Predictor. Probabilistic SVM is a suitable choice for this classification as it performs well on text data, and we are also interested in the class likelihood of the instances.

In essence, the Link Predictor works as a low-cost filter, its cost is associated with the link classifications which is very low. It also considerably prunes the search space for subsequent steps that are typically more expensive computationally.

**Language Identification.** In the second step of the bilingual site detection, the BS Detector verifies if the pages whose links were considered relevant by the Link Predictor are in the languages of interest. The motivation behind the use of language identification for our problem is, since we are interested in bilingual text, only looking at individual links of these sites might not suffice. In addition to identify the language of the pages of candidate links identified by the Link Predictor, language identification is also performed on the page that contains such links, i.e., the page that was provided as input to the BS Detector. This handles cases in which a page in a given language only contains a link to its translation in other language but not links to both versions. The language identification is then performed in all pages of that candidate list and if different pages are in the language of interest, the site is considered as bilingual. To detect the language of a given page, we use the textcat (Cavnar and Trenkle, 1994) Language Identifier.

Even though there is some cost in downloading the pages to perform this step, we show later in this section that it is only necessary to download

| Min. Likelihood | Link Predictor | | | BS Detector | | | Cost (Downloads per site) |
|---|---|---|---|---|---|---|---|
| | Rec. | Prec. | F-Meas. | Rec. | Prec. | F-Meas. | |
| 0 | 1 | 0.5 | 0.66 | 0.86 | 0.73 | 0.79 | 29.1 |
| 0.1 | 0.97 | 0.55 | 0.7 | 0.75 | 0.84 | 0.79 | 6.5 |
| 0.2 | 0.94 | 0.61 | 0.8 | 0.74 | 0.89 | 0.8 | 4.5 |
| 0.3 | 0.88 | 0.68 | 0.76 | 0.71 | 0.93 | 0.8 | 3.6 |
| 0.4 | 0.85 | 0.72 | 0.78 | 0.7 | 0.93 | 0.8 | 3.1 |
| 0.5 | 0.84 | 0.75 | 0.79 | 0.67 | 0.95 | 0.78 | 2.8 |
| 0.6 | 0.84 | 0.75 | 0.79 | 0.67 | 0.95 | 0.78 | 2.6 |
| 0.7 | 0.83 | 0.76 | 0.79 | 0.67 | 0.95 | 0.78 | 2.4 |
| 0.8 | 0.81 | 0.78 | 0.79 | 0.67 | 0.95 | 0.78 | 2.2 |
| 0.9 | 0.77 | 0.8 | 0.78 | 0.66 | 0.97 | 0.78 | 2 |

Table 1: Results obtained by the BS Detector (Link Predictor + language identification) and the Link Predictor only.

on average 2 to 3 pages per site, since the Link Predictor prunes the search space considerably.

**Evaluation.** To measure the quality of the BS Detector, we manually labeled 200 Web sites (100 positive and 100 negative) from the dmoz directory in topics related to Spanish speaking countries. A site was considered as relevant if it contained at least a pair of parallel pages. Our approach is similar to that employed by (Resnik and Smith, 2003) to label parallel text.

Since the Link Predictor outputs the likelihood of a relevant link, we varied the minimum likelihood for a link be considered as relevant. For each value, we measured its quality (precision, recall and F-measure), as well as its cost (number of downloaded pages per site in the language identification step). Table 1 presents the results for the BS Detector and the Link Predictor (first step of the BS Detector). When the minimum likelihood is 0, the language identification process checks all the links in the given pages for pairs of languages, i.e., the Link Predictor considers all the links as relevant. In this scenario, an average of 29 pages per Web site were downloaded and the recall of the BS Detector was 0.86. This implies that the language identifier was not able to detect pairs of languages in 16% of the relevant sites. As expected, the minimum likelihood is directly proportional to the precision and inversely proportional to the recall. It is interesting to note that between 0.5 and 0.8, these values do not change for the BS Detector besides the decreasing of cost. The Link Predictor shows a similar behavior. Another important observation to glean is that adding the language detection on top of the Link Predictor improves the overall precision of the bilingual site detection. For instance, when the minimum likelihood is set to 0.5, the Link Predictor's precision is 0.75 whereas that of the BS Detector is 0.95. The high precision of the BS Detector is very important to build a high-quality set of bilingual sites.

## 2.2 Crawling Policy

In this section, we focus our attention to our solution to locating bilingual sites on the Web. Previous work (Ma and Liberman, 1999) tries to perform this task by restricting the crawler in a top-level internet domain where it is supposed to contain a high concentration of these sites. For instance, Ma and Liberman (Ma and Liberman, 1999) focused the crawler in .de domain since they were interested in German/English language pairs. In this work, we do not restrict the crawler to any particular internet domain or topic. Our objective is to allow the crawler to perform a broad search while avoiding visits to unproductive Web regions.

We implemented this strategy by imposing the constraint that the crawler stays in the Web neighborhood graph of the bilingual sites that were previously discovered by the crawler. More specifically, the crawler explores the neighborhood graph defined by the bipartite graph composed by the backlink pages (BPs) of bilingual sites and the pages pointed by BPs (forward pages), see Figure 1. As we mentioned before, this strategy is based on the findings that Web communities are characterized by directed bipartite subgraphs (Kumar et al., 1999). Thus, our assumption is that the Web region comprised by this bipartite graph is rich in bilingual sites as backlink pages typically point to multiple bilingual sites. Finally, as we are looking for Web sites and not for single Web pages, the crawler only considers out-of-site links, i.e., it excludes from the bipartite graph links to internal pages of the sites.

The steps of our algorithm are shown in Algorithm 1. Initially, the user provides a set of seed URLs that are added to the frontier. The crawler then starts to download the links in the frontier. If the BS Detector identifies a page in a bilingual site, the backlinks to this page are collected and added back to the frontier. Backlink information can be retrieved through the "link:"

**Algorithm 1** Crawling Policy

```
1: Input: seeds, BS_Detector
   {seeds : seeds provided by the user, BS_Detector: the
   Bilingual Site Detector.}
2: frontier = ∅
   {Create the empty frontier.}
3: frontier.addLinks(seeds)
   {Add the seeds to the frontier.}
4: repeat
5:    link = frontier.next()
      {Retrieve from the frontier the next link to be vis-
      ited.}
6:    page = download(link)
      {Download the content of the page.}
7:    if BS_Detector.isRelevant(page) then
8:       backlinks = collectBacklins(page)
         {Collect the backlinks to the given page provided
         by a search engine API.}
9:       frontier.addLinks(backlinks)
         {Add the backlinks to the frontier.}
10:   end if
11:   if link.isBacklink() then
12:      outlinks = extractOutlinks(page)
         {Extract the outlinks of a backlink page.}
13:      frontier.addLinks(outlinks)
         {Add the outlinks to the frontier.}
14:   end if
15: until frontier.isEmpty()
```

API provided by search engines such as Google and Yahoo! (Bharat et al., 1998). In the next step, pages represented by the backlinks (backlink pages) are downloaded, their outlinks are extracted and added to the frontier. Notice that only the outlinks from the backlink pages are added to the frontier. The crawler does not explore outlinks of forward pages (forward pages are pages pointed by backlink pages, see Figure 1).

### 2.3 Forward-Link and Backlink Classifiers

Retaining the crawler in the graph neighborhood of bilingual sites (the bipartite graph) is our first attempt towards an effective search for such sites. However, there may be many links in the graph that do not lead to relevant sites. In order to identify promising URLs in the two different page sets of the bipartite graph, we employ supervised learning. For each set (backlink and forward sets), the crawler builds a classifier that outputs the relevance of a given link in that particular set. Relevant links in the forward pages' set represent URLs of bilingual sites, i.e., links that give immediate benefit, whereas relevant links in the backlink pages' set are URLs of backlink pages that contain outlinks to bilingual sites (delayed benefit).

Previous approaches for focused crawling (Chakrabarti et al., 2002; Rennie and McCallum, 1999; Barbosa and Freire, 2007) also use patterns on links to prioritize them. But instead of using link classifiers specialized in different link layers, they build a single classifier. The advantage of having multiple classifiers is that it decomposes a complex problem into simpler subproblems in which each classifier is dedicated to a subset of more homogeneous hypothesis (Gangaputra and Geman, 2006). Diligenti et al. (Diligenti et al., 2000) also proposed the use of multiple classifiers to guide the crawler. But instead of looking at link patterns, they use the content of the pages.

In summary, the Forward-Link Classifier predicts the most promising links for the forward pages, whereas the Backlink Classifier identifies the most promising links for the backlink pages. Both classifiers use as features the neighborhood of links. The link neighborhood is composed by four contextual categories: URL (without the host), host, anchor, and text around the link. Since the number of extracted features tends to be large (and most of them have very low frequency), we remove stop-words and stem the remaining words. Note that features are associated with a context. For example, if the word "hotel" appears both in the URL and anchor text of a link, it is added as a feature in both contexts. It is important to note that words in the host context have an important role, since many parallel corpus sites are in a country's internet domain, e.g., es, de, etc. In fact, as we mentioned before, some previous approaches (Ma and Liberman, 1999) restrict the crawl within these domains to collect parallel data. But instead of pre-defining a set of domains, the crawler in our work automatically identifies the most important ones during its crawling process.

As one can expect, the two classifiers perform a different role. For the Backlink Classifier, features such as "link" and "directory" in the URL obtained have high frequency in the training data. These words usually occur in the URL of pages that point to many different sites, e.g., http://www.rentaccomspain.com/links.asp. The Forward-Link Classifier is more focused on topics. Words such as "hotel", "air", "art" and "language" were some of the frequent features used by it.

The two classifiers are automatically created during the crawling process. Initially, the crawler starts with no link prioritization. After a specified number of crawled pages, a learning iteration

is performed by collecting the link neighborhood of the links that point to relevant and non-relevant pages in each set. The result of this process is used as training data for the Backlink and Forward-Link classifiers. Similar to previous focused crawling approaches (Chakrabarti et al., 2002; Barbosa and Freire, 2007), we use Naive Bayes algorithm for this purpose. As a final step, the relevance of the links in the frontier are updated based on the new classifiers.

## 3 Crawling Experiments

In this section, we assess our crawling strategy to locate bilingual sites and compare it with other crawling approaches.

### 3.1 Experimental Setup

**Crawling Strategies.** We executed the following crawling strategies to locate bilingual sites:

- Forward Crawler (FC): The forward crawler randomly follows the forward links without any restriction;
- Focused Crawler (FocC): although our strategy does not restrict its search to a particular domain, we set up a focused crawler (Chakrabarti et al., 2002) in the travel domain for comparison. The focused crawler is composed by a page classifier that restricts the crawl to pages in the travel domain and a link classifier that guides the crawler's link visitation to avoid unproductive Web regions (see (Chakrabarti et al., 2002) for more details);
- Out-of-site Back/Forward Crawler (OBFC): The out-of-site back/forward crawler uses the crawling strategy proposed in this paper without any prioritization to the links;
- Classifier-Based Out-of-site Back/Forward Crawler (COBFC): The classifier-based out-of-site back/forward is the OBFC along with the Backlink and Forward-link classifiers to prioritize the links in the frontier. Both classifiers are created after crawling 20,000 pages.

We set up the crawlers to locate bilingual sites in English and Spanish. Each configuration collected 100,000 pages and 1,000 links were provided as seeds. These were randomly selected from the URLs available on the Open Directory Project[1] related to Spanish speaking countries.

[1] http://www.dmoz.org/

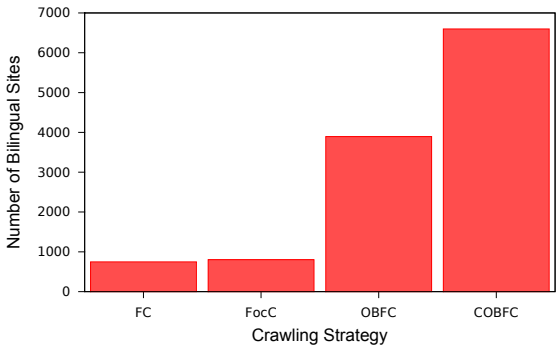


Figure 3: Total of bilingual sites collected by the crawling strategies in a crawl of 100,000 pages.

**Effectiveness measure.** The performance of the crawling strategies was measured by the total number of bilingual sites collected after the bilingual site detection during the crawl. The minimum likelihood used by BS Detector to consider a link as relevant was 0.8 since we are interested in obtain a high-quality collection of bilingual sites (see Section 2.1).

### 3.2 Assessing the Bilingual Site Crawler

In Figure 3, we present the total of bilingual sites collected by each crawling configuration after collecting 100,000 pages. Our crawling strategy, COBFC, collected the greatest number of bilingual sites (6598 sites). This result empirically confirms that our approach of restricting the crawler to the neighborhood of bilingual sites by using back and forward links, along with classifiers that prioritize these links is in fact effective for locating bilingual sites.

The comparison between the top two strategies, namely, COBFC (6598 bilingual sites) and OBFC (3894 bilingual sites) shows that: (1) the Backlink and Forward-link classifiers used to prioritize the links in the frontier improve the crawler's performance; and (2) even with no link prioritization, our strategy of restricting the search to the bipartite graph of backlink and forward pages is able to obtain good results. We can conclude from these numbers that bilingual sites are close to each other when one considers their backlinks. As we mentioned previously, this can be attributed to the fact that backlinks are typically hubs to bilingual sites.

From the experimental results, it is clear that our crawling is effective for locating bilingual sites on the Web. The main limitation, however, is that it relies on an external component (search engine) to provide backlinks. In the experiments presented

in this work, the use of a search engine slowed down the crawling execution since we did not want to submit many requests to the search engine and consequently have the backlink requests halted.

A final note regarding our crawling strategy is that even though we do not restrict it to any particular topic, as the crawling process evolves, it automatically focuses on topics where there is a higher concentration of parallel data, as travel, translator sites, etc. This is different from conventional approaches that explicitly constrain the crawl based on topics.

## 4 Machine Translation Experiments

In this section, we exploit the parallel text obtained through our crawling strategy as augmented data in machine translation. We use a phrase-based statistical machine translation system (Koehn et al., 2007) in all the experiments.

### 4.1 Experimental Setup

**Web data.** We focus on English and Spanish as the bilingual pair of languages. We used the crawling strategy presented in the previous section to obtain a set of 20186 bilingual sites. The parallel text from these sites was mined using the technique presented in (Rangarajan et al., 2011). A total of initial 4.84M bilingual sentence pairs were obtained from this process. We used length-based and word-based filters as well as a language model to filter these initial sentence pairs. After cleanup, a total of 2,039,272 bilingual sentence pairs was obtained from the crawling data.

**Development and Test Data.** In order to obtain a representative reference development and test set, we manually created bilingual sentences in the hospitality and tourism domain. A bilingual speaker was given instructions to create dialogs in a variety of travel scenarios such as *making a hotel reservation, booking a taxi, checking into a hotel, calling front desk and reporting problems*, etc. A total of 49 scenarios were created that resulted in 1019 sentences, 472 of which was used for development and 547 for testing. The dialogs were created in English and then translated to Spanish. The development and test sets are not very large, mainly because creating high quality bilingual data for a particular domain is expensive. We have given due consideration to create a test set that is highly similar to the domain of operation. We are working on evaluating the performance of the crawled data on different domains as part of future work (as we translate more data through human annotations).

**MT Models.** We performed machine translation experiments in both directions, English-Spanish and Spanish-English. Europarl data is the only source of parallel data (English/Spanish) that we have access to, and hence it serves as the data for our baseline translation model. Although the model can be considered to be out-of-domain with respect to our test domain, its language style is more similar to our test set (spoken dialogs) in comparison with the Web data. The Europarl data comprised 1.48M bilingual sentence pairs.

The Web data translation model was trained on the sentences resulting from the Web crawler. We also used a combination of the two models that we call as combined model. The combined model uses both the phrase tables during decoding. The reordering table was also concatenated from the two models.

### 4.2 Results

Table 2 presents the translation performance in terms of various metrics such as BLEU (Papineni et al., 2002), METEOR (Lavie and Denkowski, 2010) and Translation Edit Rate (TER) (Snover et al., 2006). The language model was a 5 gram language model optimized on the development set based on perplexity and the translation weights of the log-linear model were learned using Minimum Error Rate Training.

While the out-of-domain model trained using Europarl data achieves a BLEU score of 20.65 on the test set (tourism and hospitality domain) for English-Spanish, the model constructed by augmenting the web crawling data to europarl data achieves a relative improvement of 22%. Similar improvements hold for Spanish-English translation. The METEOR scores reported in Table 2 were computed only for exact match (synonyms and stemmed matches were not considered). For all three objective metrics, we achieve significant improvements in translation performance. The results demonstrate the efficacy of our bilingual crawling approach for harvesting parallel text for machine translation. The bilingual crawler can be initialized with a different policy based on the test domain of interest and hence our scheme is generalizable.

Regarding the results of Europarl versus Web data alone, the reason for the lower translation

| Model | Training data | English-Spanish | | | Spanish-English | | |
|---|---|---|---|---|---|---|---|
| | | BLEU | METEOR | TER | BLEU | METEOR | TER |
| Baseline | Europarl | 20.65 | 16.76 | 71.52 | 25.26 | 41.09 | 64.76 |
| | Web | 16.94 | 14.81 | 79.28 | 23.48 | 39.65 | 66.11 |
| | Combined | 23.00 | 17.90 | 68.86 | 28.86 | 44.18 | 61.24 |

Table 2: Automatic evaluation metric scores for translation models from out-of-domain data, Web data and combined models.

quality of the web crawled data on the test set considered in the experiments is mainly due to the style of the test set. Even though the domain of the crawler is travel and hospitality, the sentences in the test set are more conversational and better matched with Europarl in terms of BLEU metric. On the other hand, the METEOR metric that accounts for the overlapping unigrams is much closer for Europarl and Web data, i.e., the vocabulary coverage is comparable.

## 5 Related Work

There are basically two main types of approaches to locate parallel corpora: query-based (Resnik and Smith, 2003; Resnik, 1998; Chen and Nie, 2000; Tomás et al., 2005) and crawling-based (Ma and Liberman, 1999; Chen et al., 2004).

Query-based approaches typically try to explore common patterns that occur in this kind of data by using them as search queries. For instance, STRAND (Resnik and Smith, 2003; Resnik, 1998) tries to locate candidate parallel pages by issuing queries like: (anchor:“english” OR anchor:“anglais”) AND (anchor:“french” OR anchor:“francais”). Chen and Nie (Chen and Nie, 2000) used a similar principle to obtain two sets of candidate sites by issuing queries as anchor:“english version” to a search engine, and then taking the union. More recently, Hong et al. (Hong et al., 2010) proposed a method that discovers document pairs by first selecting the top words in a source language document, translating these words and issuing them as a query to a search engine. The main limitation of these previous approaches is that they only rely on the search engine results to obtain the parallel pages. And, since search engines restrict the total number of results per query and the number of requests, there is a limitation in terms of the total number of sites that can be collected. This is confirmed by the numbers presented in their experimental evaluation. For instance, Chen and Nie (Chen and Nie, 2000) reported a total of only 185 candidate sites for English-Chinese corpora.

With respect to crawling-based approaches for locating parallel text, there is not much prior work in this area. In fact, most of the research in this area is focused more on the problem of identifying the text pairs (Munteanu and Marcu, 2005; Zhang et al., 2006; Uszkoreit et al., 2010) than actually locating them. They typically use simple strategies to locate parallel text without exploring the Web link structure. For example, Ma and Liberman (Ma and Liberman, 1999) try to achieve this goal by simply restricting the crawler within in a particular internet domain whereby there might be a good chance of finding this kind of data.

## 6 Conclusions

This paper presents a novel focused crawling strategy to locate bilingual sites. It keeps its search in the bipartite graph composed by the backlink pages of bilingual sites (already discovered by the crawler) and the pages pointed by them. To focus on the most promising regions in this graph, the crawler explores the patterns presented in its links to guide its visitation policy. Another novelty proposed in this paper is our low-cost and high-precision strategy to detect a bilingual site. It performs this task in two steps. First, it relies on common patterns found in the internal links of these sites to compose a classifier that identifies link pages as entry points to parallel data in these sites. Second, it verifies whether these pages are in fact in the languages of interest. Our experiments showed that our crawling strategy is more effective in finding bilingual sites than the baseline approaches and that our bilingual site detection has high-precision while being efficient. We also demonstrated the efficacy of our crawling approach by performing machine translation experiments using the parallel text obtained from the bilingual sites identified by the crawler.

An interesting venue to pursue in a future work is to verify whether the crawling strategy proposed in this paper also works in other types of domains where regular focused crawling may have issues in finding the targeted Web sources.

## References


D. Ahlers and S. Boll. 2009. Adaptive geospatially focused crawling. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 445–454.

L. Barbosa and J. Freire. 2007. An adaptive crawler for locating hidden-web entry points. In *WWW*, pages 441–450.

K. Bharat, A. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. 1998. The connectivity server: Fast access to linkage information on the web. *Computer Networks and ISDN Systems*, 30(1-7):469–477.

W.B. Cavnar and J.M. Trenkle. 1994. N-gram-based text categorization. pages 161–175.

S. Chakrabarti, M. Berg, and B. Dom. 1999. Focused crawling: A new approach to topic-specific web resource discovery. *Computer Networks*, 31(11-16):1623–1640.

S. Chakrabarti, K. Punera, and M. Subramanyam. 2002. Accelerated focused crawling through online relevance feedback. In *WWW*, pages 148–159.

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

J. Chen, R. Chau, and C.H. Yeh. 2004. Discovering parallel text from the World Wide Web. In *workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 161–165.

M. Diligenti, F. Coetzee, S. Lawrence, C. Lee Giles, and M. Gori. 2000. Focused Crawling Using Context Graphs. In *VLDB*, pages 527–534.

S. Gangaputra and D. Geman. 2006. A design principle for coarse-to-fine classification. In *Computer Vision and Pattern Recognition*, volume 2, pages 1877–1884.

D.A. Grossman and O. Frieder. 2004. *Information retrieval: Algorithms and heuristics*. Kluwer Academic Pub.

G. Hong, C. Li, M. Zhou, and H. Rim. 2010. An empirical study on web mining of parallel data. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 474–482, Stroudsburg, PA, USA. Association for Computational Linguistics.

L. Jiang, S. Yang, M. Zhou, X. Liu, and Q. Zhu. 2009. Mining bilingual data from the web with adaptively learnt patterns. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL*, pages 870–878.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of ACL*, pages 177–180.

P. Koehn. 2005a. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5.

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

R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. 1999. Trawling the Web for emerging cyber-communities. *Computer networks*, 31(11-16):1481–1493.

A. Lavie and M. Denkowski. 2010. The METEOR metric for automatic evaluation of machine translation. *Machine Translation*.

X. Ma and M. Liberman. 1999. Bits: A method for bilingual text search over the web. In *Machine Translation Summit VII*.

I.D. Melamed. 2001. *Empirical methods for exploiting parallel texts*. MIT Press.

D. S. Munteanu and D. Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31:477–504, December.

C. Olston and M. Najork. 2010. Web Crawling. *Information Retrieval*, 4(3):175–246.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.

J. C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. pages 61–74.

V. K. S. Rangarajan, L. Barbosa, and S. Bangalore. 2011. A Scalable Approach for Building a Parallel Corpus from the Web. In *Proceedings of 12th Annual Conference of the International Speech Communication Association*.

J. Rennie and A. McCallum. 1999. Using Reinforcement Learning to Spider the Web Efficiently. In *ICML*, pages 335–343.

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

P. Resnik. 1998. Parallel strands: A preliminary investigation into mining the web for bilingual text. *Machine Translation and the Information Soup*, pages 72–82.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

J. Tomás, E. Sánchez-Villamil, L. Lloret, and F. Casacuberta. 2005. WebMining: An unsupervised parallel corpora web retrieval system. In *Proceedings from the Corpus Linguistics Conference*.

J. Uszkoreit, J. M. Ponte, A. C. Popat, and M. Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1101–1109, Stroudsburg, PA, USA. Association for Computational Linguistics.

Y. Zhang, K. Wu, J. Gao, and P. Vines. 2006. Automatic Acquisition of Chinese–English Parallel Corpus from the Web. *Advances in Information Retrieval*, pages 420–431.