A modular open-source focused crawler for mining monolingual and bilingual corpora from the web

Vassilis Papavassiliou **Prokopis Prokopidis**
Institute for Language and Speech Processing
Athena Research and Innovation Center
Athens, Greece
{vpapa, prokopis}@ilsp.gr

Gregor Thurmair
Linguatec
Gottfried-Keller-Str. 12, 81245
Munich, Germany
g.thurmair@linguatec.de

Abstract

This paper discusses a modular and open-source focused crawler (ILSP-FC) for the automatic acquisition of domain-specific monolingual and bilingual corpora from the Web. Besides describing the main modules integrated in the crawler (dealing with page fetching, normalization, cleaning, text classification, de-duplication and document pair detection), we evaluate several of the system functionalities in an experiment for the acquisition of pairs of parallel documents in German and Italian for the "Health & Safety at work" domain.

1 Introduction and motivation

There is a growing literature on using the Web for constructing various types of text collections, including monolingual, comparable, parallel and/or domain-specific corpora. Such resources can be used by linguists studying language use and change (Kilgarriff and Grefenstette, 2003), and at the same time they can be exploited in applied research fields like machine translation and multilingual information extraction. Moreover, these collections of raw data can be automatically annotated and used to produce, by means of induction tools, a second order or synthesized derivatives: rich lexica (with morphological, syntactic and lexico-semantic information), large bilingual dictionaries (word and multiword based) and transfer grammars.

To this end, several tools (i.e. web crawlers, HTML parsers, language identifiers, HTML cleaners, etc.) have been developed and combined in order to produce corpora useful for specific tasks. However, to the best of our knowledge, most of the available systems either omit some processing tasks or require access to the results of a search engine. For instance, the BootCaT toolkit (Baroni et al., 2006), a well-known suite of Perl scripts for bootstrapping specialized language corpora from the web, uses the Bing search engine and allows up to 5,000 queries per month.

In this paper, we present ILSP-FC, a modular system that includes components and methods for all the tasks required to acquire domain-specific corpora from the Web. The system is available as an open-source Java project[1] and due to its modular architecture, each of its components can be easily substituted by alternatives with the same functionalities. Depending on user-defined configuration, the crawler employs processing workflows for the creation of either monolingual or bilingual collections. For users wishing to try the system before downloading it, two web services[2] allow them to experiment with different configuration settings for the construction of monolingual and bilingual domain-specific corpora.

The organization of the rest of the paper is as follows. In Section 2, we refer to recent related work. In Section 3, we describe in detail the workflow of the proposed system. A solution for bootstrapping the focused crawler input is presented in Section 4. Then, an experiment on acquiring parallel documents in German and Italian for the "Health & Safety at work" domain (H&S) is described in Section 5, which also includes evaluation results on a set of criteria including parallelness and domain specificity. We conclude and mention future work in Section 6.

2 Related work

Web crawling for building domain-specific monolingual and/or parallel data involves several tasks (e.g. link ranking, cleaning, text classification, near-duplicates removal) that remain open issues. Even though there are several proposed methods

[1] http://nlp.ilsp.gr/redmine/projects/ilsp-fc

[2] http://nlp.ilsp.gr/ws/

Proceedings of the 6th Workshop on Building and Using Comparable Corpora, pages 43–51,
Sofia, Bulgaria, August 8, 2013.

for each of these tasks, in this section we refer only to a few indicative approaches.

Olston and Najork (2010) outline the fundamental challenges and describe the state-of-the-art models and solutions for web crawling. A general framework to fairly evaluate focused crawling algorithms under a number of performance metrics is proposed by Srinivasan et al. (2005). A short overview of cleaning methods is presented in Spousta et al. (2008) and the comparison of such methods is discussed in Baroni et al. (2008). Several algorithms (Qi and Davison, 2009) exploit the main content and the HTML tags of a web page in order to classify a page as relevant to a targeted domain or not. Methods for the detection and removal of near-duplicates (i.e. acquired web pages that have almost the same content) are reviewed and compared in Theobald et al. (2008).

Efficient focused web crawlers can be built by adapting existing open-source frameworks like Heritrix[3], Nutch[4] and Bixo[5]. For instance, Combine[6] is an open-source focused crawler that is based on a combination of a general web crawler and a text classifier. Other approaches make use of search engines APIs to identify in-domain web pages (Hong et al., 2010) or multilingual web sites (Resnik and Smith, 2003). Starting from these pages, Almeida and Simões (2010) try to detect which links point to translations, while Shi et al. (2006) harvest multilingual web sites and extract parallel content from them. Bitextor (Esplà-Gomis and Forcada, 2010) combines language identification with shallow features that represent HTML structures to mine parallel pages.

Besides structure similarity, systems like PTMiner (Nie et al., 1999) and WeBiText (Désilets et al., 2008) filtered fetched web pages by keeping only those containing language markers in their URLs. Chen et al. (2004) proposed the Parallel Text Identification System, which incorporated a content analysis module using a predefined bilingual wordlist. Similarly, Zhang et al. (2006) and Utiyama et al. (2009) adopted the use of aligners in order to estimate the content similarity of candidate parallel web pages or mixed languages pages. Barbosa et al. (2012) proposed the use of bilingual dictionaries and generated translations (e.g. by Google Translate and Microsoft Bing) to extract parallel content from multilingual sites.

[3]http://crawler.archive.org/
[4]http://nutch.apache.org
[5]http://openbixo.org/
[6]http://combine.it.lth.se/

3 System architecture

In this section, we describe the main modules integrated in ILSP-FC. In general, the crawler initializes its frontier (i.e. the list of pages to be visited) from a seed URL list provided by the user (or constructed semi-automatically, see Section 4), classifies fetched pages as relevant to the targeted domain, extracts links from fetched web pages and adds them to the list of pages to be visited.

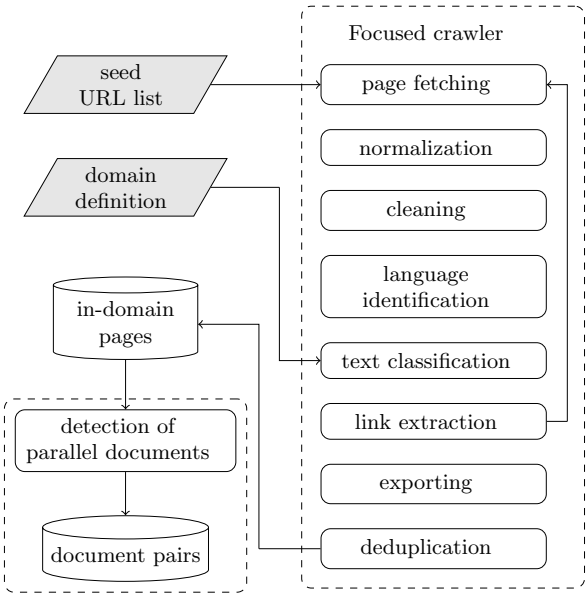

Figure 1: System architecture

In order to ensure modularity and scalability, the crawler is built using Bixo, an open source web mining toolkit that allows easy configuration of workflows and runs on top of the Hadoop[7] framework for distributed data processing.

3.1 Page Fetcher

The first module concerns page fetching. A multithreaded crawling implementation has been adopted in order to ensure concurrent visiting of multiple hosts. Users can configure several settings that determine the fetching process, including number of concurrent harvesters and filtering out specific document types. The crawler always respects standard robots.txt files, while politeness can also be affected with the use of settings regarding time intervals for revisiting URLs from the same website, maximum number of URLs from a specific host per iteration, maximum number of attempts to fetch a web page etc.

[7]http://hadoop.apache.org

3.2 Normalizer

The normalizer module uses the Apache Tika toolkit [8] to parse the structure of each fetched web page and extract its metadata. Extracted metadata are exported at a later stage (see Subsection 3.7) if the web document is considered relevant to the domain. The text encoding of the web page is also detected based on the HTTP Content-Encoding header and the charset part of the Content-Type header, and if needed, the content is converted into UTF-8. Besides default conversion, special care is taken for normalization of specific characters like no break space, narrow no-break space, three-per-em space, etc.

3.3 Cleaner

Apart from its textual content, a typical web page also contains boilerplate, i.e. "noisy" elements like navigation headers, advertisements, disclaimers, etc., which are of only limited or no use for the production of good-quality language resources. For removing boileplate, we use a modified version of Boilerpipe [9] (Kohlschütter et al, 2010) that also extracts structural information like *title*, *heading* and *list item*. At this stage, text is also segmented in paragraphs on the basis of specific HTML tags like <*p*>, <*br*> and <*li*>. Paragraphs judged to be boilerplate and/or detected as titles, etc. are properly annotated (see Subsection 3.7).

3.4 Language Identifier

The next processing module deals with language identification. We use the Cybozu[10] language identification library that considers n-grams as features and exploits a Naive Bayes classifier for language identification. If a web page is not in the targeted language, its only further use is in extraction of new links. Even though the main content of a web page is in the targeted language, it is likely that the web page includes a few paragraphs that are not in this language.Thus, the language identifier is also applied on each paragraph and marks them properly (see Subsection 3.7).

3.5 Text Classifier

The aim of this module is to identify if a page that is normalized and in the targeted language contains data relevant to the targeted domain. To this end, the content of the page is compared to a user-provided domain definition. Following the string-matching method adopted by the Combine web crawler, the definition consists of term triplets (<relevance weight, (multi-word) term, subdomain>) that describe a domain and, optionally, subcategories of this domain. Language-dependent stemmers from the Lucene[11] project are used to stem user-provided terms and document content. Based on the number of terms' occurrences, their location in the web page and the weights of found terms, a page relevance score p is calculated as follows:

$$p = \sum_{i=1}^{N} \sum_{j=1}^{4} n_{ij} \cdot w_i^t \cdot w_j^l,$$

where N is the amount of terms in the domain definition, w_i^t is the weight of term i, w_j^l is the weight of location j and n_{ij} denotes the number of occurrences of term i in location j. The four discrete locations in a web page are *title*, *metadata*, *keywords*, and plain text, with respective weights of 10, 4, 2, and 1.

Moreover, the amount of unique domain terms found in the main content of the page, m, is calculated. Then, the values p and m are compared with two predefined thresholds (t_1 and t_2) and if both values are higher than the thresholds, the web page is categorized as relevant to the domain and stored. It is worth mentioning that the user can affect the strictness of the classifier by setting the values of both thresholds in the crawler's configuration file.

3.6 Link Extractor

Even when a web page is not stored (because it was deemed irrelevant to the domain, or not in the targeted language), its links are extracted and added to the list of links scheduled to be visited. Since the crawling strategy is a critical issue for a focused crawler, the links should be ranked and the most promising links (i.e. links that point to "in-domain" web pages or candidate translations) should be followed first. To this end, a score s_l is calculated for each link l as follows:

$$s_l = c + p/L + \sum_{i=1}^{N} n_i \cdot w_i$$

where L is the amount of links originating from the source page, N is the amount of terms in the domain definition, n_i denotes the number of occurrences of the i-th term in the link's surrounding text and w_i is the weight of the i-th term. By using this

[8] http://tika.apache.org

[9] http://code.google.com/p/boilerpipe/

[10] http://code.google.com/p/language-detection/

[11] http://lucene.apache.org/

formulation, the score link is mainly influenced by the "domainess" of its surrounding text.

The parameter c is only added in case the crawler is used for building bilingual collections. It gets a high positive value if the link under consideration originates from a web page in L1 and "points" to a web page that is probably in L2. This is the case when, for example, L2 is German and the anchor text contains strings like "de", "Deutsch", etc. The insertion of this parameter forces the crawler to visit candidate translations before following other links.

3.7 Exporter

The Exporter module generates an XML file for each stored web document. Each file contains metadata (e.g. language, domain, URL, etc.) about the corresponding document inside a header element. Moreover, a <*body*> element contains the content of the document segmented in paragraphs. Apart from normalized text, each paragraph element <*p*> is enriched with attributes providing more information about the process outcome. Specifically, <*p*> elements in the XML files may contain the following attributes: i) *crawlinfo* with possible values *boilerplate*, meaning that the paragraph has been considered boilerplate (see Subsection 3.3), or *ooi-lang*, meaning that the paragraph is not in the targeted language; ii) *type* with possible values: *title, heading* and *listitem*; and iii) *topic* with a string value including all terms from the domain definition detected in this paragraph.

3.8 De-duplicator

Ignoring the fact[12] that the web contains many near-duplicate documents could have a negative effect in creating a representative corpus. Thus, the crawler includes a de-duplicator module that represents each document as a list containing the MD5 hashes of the main content's paragraphs, i.e. paragraphs without the *crawlinfo* attribute. Each document list is checked against all other document lists, and for each candidate pair, the intersection of the lists is calculated. If the ratio of the intersection cardinality to the cardinality of the shortest list is more than 0.8, the documents are considered near-duplicates and the shortest is discarded.

[12] Baroni et al. (2009) reported that during building of the Wacky corpora, the amount of collected documents was reduced by more than 50% after de-duplication.

3.9 Pair Detector

After in-domain pages are downloaded, the Pair Detector module uses two complementary methods to identify pairs of pages that could be considered parallel. The first method is based on co-occurrences, in two documents, of images with the same filename, while the second takes into account structural similarity.

In order to explain the workflow of the pair detection module, we will use the multilingual website `http://www.suva.ch` as a running example. Crawling this website using the processes described in previous subsections provides a pool of 707 HTML files (and their exported XML counterparts) that are found relevant to the H&S domain and in the targeted DE and IT languages (376 and 331 files, respectively).

Each XML file is parsed and the following features are extracted: i) the document *language*; ii) the *depth* of the original source page, (e.g. for `http://domain.org/d1/d2/d3/page.html`, depth is 4); iii) the *amount of paragraphs*; iv) the *length* (in terms of tokens) of the clean text; and v) the *fingerprint* of the main content, which is a sequence of integers that represent the structural information of the page, in a way similar to the approach described by Esplà-Gomis and Forcada (2010). For instance, the *fingerprint* of the extract in Figure 2 is [-2, 28, 145, -4, 9, -3, 48, -5, 740] with *boilerplate* paragraphs ignored; -2, -3 and -4 denote that the *type* attributes of corresponding <*p*> elements have *title*, *heading* and *listitem* values, respectively; -5 denotes the existence of the *topic* attribute in the last <*p*>; and positive integers are paragraph lengths in characters.

The *language* feature is used to filter out pairs of files that are in the same language. Pages that have a depth difference above 1 are also filtered out, on the assumption that it is very likely that translations are found at the same or neighbouring depths of the web site tree.

Next, we extract the filenames of the images from HTML source and each document is represented as a list of image filenames. Since it is very likely that some images appear in many web pages, we count the occurrence frequency of each image and discard relatively frequent images (i.e. Facebook and Twitter icons, logos etc.) from the lists.

In order to classify images into "critical" or "common" (see Figure 3) we need to calculate a threshold. In principle, one should ex-

```
<p type="title">Strategia degli investimenti</p> <!-- -2, 28-->
<p >I ricavi degli investimenti sono un elemento essenziale per finanziare le
    rendite e mantenere il potere d'acquisto dei beneficiari delle rendite.</p>
    <!--145-->
<p type="listitem">Document:</p> <!-- -4, 9 -->
<p crawlinfo="boilerplate" type="listitem">Factsheet "La strategia d'investimento
    della Suva in sintesi" (Il link viene aperto in una nuova finestra) </p> <!--
    ignored -->
<p type="heading">Perché la Suva effettua investimenti finanziari?</p> <!-- -3,
    48-->
<p topic="prevenzione degli infortuni;infortunio sul lavoro">Nonostante i molti
    sforzi compiuti nella prevenzione degli infortuni sul lavoro e nel tempo libero
    ogni anno accadono oltre 2500 infortuni con conseguenze invalidanti o mortali.
    In questi casi si versa una rendita per invalidità agli infortunati oppure una
    rendita per orfani o vedovile ai superstiti. Nello stesso anno in cui
    attribuisce una rendita, la Suva provvede ad accantonare i mezzi necessari a
    pagare le rendite future. La maggior parte del patrimonio investito dalla Suva è
     rappresentato proprio da questi mezzi, ossia dal capitale di copertura delle
    rendite. La restante parte del patrimonio è costituta da accantonamenti per
    prestazioni assicurative a breve termine come le spese di cura, le indennità
    giornaliere e le riserve.</p> <!-- -5, 740-->
```

Figure 2: An extract of an XML file for an Italian web page relevant to the H&S domain.

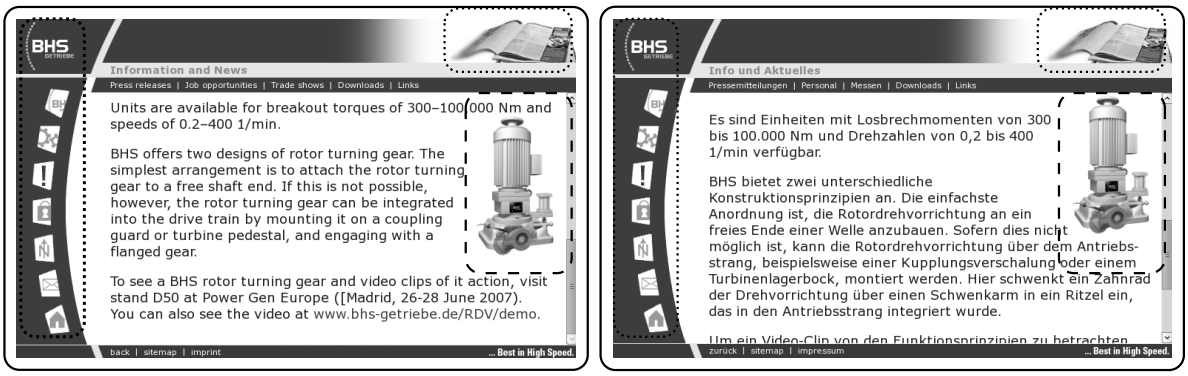

Figure 3: Critical (dashed) and common (dotted) images in a multilingual (EN/DE) site.

pect that low/high frequencies correspond to "critical"/"common" images. We employ a non-parametric approach for estimating the probability density function (Alpaydin, 2010) of the image frequencies using the following formula:

$$\hat{p}(x) = \frac{1}{Mh} \sum_{t=1}^{M} K(\frac{x-x^t}{h})$$

where the random variable x defines the positions (i.e. images frequencies) at which the $\hat{p}(x)$ will be estimated, M is the amount of images, x^t denotes the values of data samples in the region of width h around the variable x, and $K(\cdot)$ is the normal kernel that defines the influence of values x^t in the estimation of $\hat{p}(x)$. The optimal value for h, the optimal bandwidth of the kernel smoothing window, was calculated as described in Bowman and Azzalini (1997).

Figure 4 illustrates the normalized histogram of image frequencies in the example collection and the estimated probability density function. One can identify a main lobe in the low values, which corresponds to "critical" images. Thus, the threshold is chosen to be equal to the minimum just after this lobe. The underlining assumption is that if a web page in L1 contains image(s) then the web page with its translation in L2 will contain more or less the same images. In case this assumption is not valid for a multilingual site (i.e. there are only images that appear in all pages, e.g. template icons), probably all images will be included. To eliminate this, we discard images that exist in more than 10% of the total HTML files.

Following this step, each document is examined against all others and two documents are considered parallel if a) the ratio of their paragraph amounts (the ratio of their lengths in terms of para-

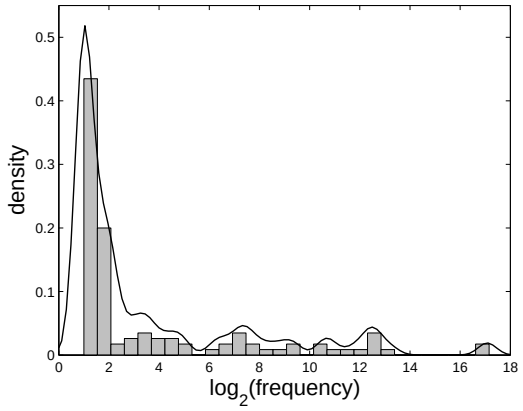

Figure 4: The normalized histogram and the estimated pdf of the image frequencies.

graphs), b) the ratio of their clean text lengths (in terms of tokens), and c) the Jaccard similarity coefficient of their image lists, are higher than empirically predefined thresholds.

More pairs are detected by examining structure similarity. Since the XML files contain information about structure, content (i.e. titles, headings, list items) and domain specificity (i.e. paragraphs with the *topic* attribute), we use these files instead of examining the similarity of the HTML source. A 3-dimensional feature vector is constructed for each candidate pair of parallel documents. The first element in this vector is the ratio of their fingerprint lengths, the second is the ratio of their sizes in paragraphs, and the third is the ratio of the edit distance of the fingerprints of the two documents to the maximum fingerprint length. Classification of a pair as parallel is achieved using a soft-margin polynomial Support Vector Machine trained with the positive and negative examples collected during our previous work (Pecina et al., 2012). Note that the dataset included only candidate pairs that met the criteria regarding the ratio of paragraphs amounts and the ratio of text lengths, mentioned above. As a result, negative instances (i.e. pairs of documents that have similar structure but are not real pairs) did not heavily outnumber positive ones and thus the training was not imbalanced (Akbani et al., 2004).

4 Bootstrapping the input of the focused crawler

In the work presented in previous sections, we assumed that users had access to already existing lists of seed terms and URLs for the initialization of the frontier and the classifier. But what if manually compiled resources for a particular domain/language(s) combination (e.g. ES/FR terminology for endocrinology or lists of EN/DE web sites related to floriculture) are impossible or difficult to find? Can we bootstrap such resources and provide them to users for post-editing? In this section, we present ongoing work towards this goal using the category graph and the external links of multilingual editions of Wikipedia.

We initialize the bootstrapping process by searching for a term defining the domain of interest (e.g. "ballet", "automotive accessories") in the category graph of the EN wikipedia. If a category is found, we recursively collect all pages in this category and its subcategories for a predefined depth. For each page we extract its title and we consider it a term that can participate in a list of domain-related seed terms. We use a set of pattern matching rules that exclude certain titles like those of disambiguation and redirect pages. Other rules exclude titles that refer to lists of related pages or titles that use upper case or title case and are probably abbreviations and named entities, respectively. Obviously, in a different setting where, for example, a user is interested in discovering named entities related to a domain, these titles should be handled differently.

The next step involves utilizing the links from each EN page to articles in wikipedias written in other languages. Based on which languages we are interested in, we again consider each title a seed term in language LANG, this time also storing the information that the term is also a LANG translation of the EN term.

During traversing the EN category graph and visiting corresponding articles in other languages, we also populate a list of seed URLs for the focused crawler, by keeping record of all links to URLs outside `wikipedia.org`. At this stage, we have all necessary resources to initiate monolingual focused crawls in each language we are interested in.

An optional last stage targets the automatic discovery of sites with multilingual content where parallel documents can be extracted from. During this stage, we visit each of the external links we collected and detect the language of the web page this link points to. From this web page, we extract its links and examine whether the anchor text of each link matches a set of patterns indicating that

this link points to a translation (in a way similar to the process described in Subsection 3.6). If translation links are found, we store the site as a candidate for bilingual focused crawling. Also, since it is common that links to multilingual editions of a web site are not present in all of its pages, we repeat the same process for the home page of the site. Notice that it is a task for the FC to detect whether these sites (or one of their sections) also contain parallel documents in the targeted domain.

In a first set of experiments following this approach, we used September 2012 snapshots[13] for English, French, German, Greek, Portuguese and Spanish wikipedias (EN, FR, DE, EL, PT and ES, respectively). Although we leave detailed evaluation of created resources for future work, we present as example output a list of terms related to "Flowers" in Table 1. Notice that, since the number of articles of multilingual wikipedias varies considerably, the term list extracted for languages like EL is, as expected, smaller compared, for example, to the 547 and 293 terms collected for EN and ES, respectively. Finally, using the URLs extracted from the articles on the "Flowers" domain, Table 2 contains a sample of web sites detected for containing relevant multilingual content.

5 Evaluation Results

In order to assess the quality of the resources that ILSP-FC can produce, we evaluated it in a task of acquiring pairs of parallel documents in German and Italian for the "Health & Safety at work" (Arbeitsschutz/Sicurezza sul lavoro) domain. We assume that this task is relatively difficult, i.e. that the number of documents in this domain and pair of languages is relatively small in the web. Overall, our system delivered 807 document pairs for H&S, containing 1.40 and 1.21 million tokens for IT and DE, respectively. Numbers refer to tokens in the main content of the acquired web pages, i.e. to tokens in paragraphs without the attribute *crawlinfo* (see Subsection 3.7).

A sample of the acquired corpora were evaluated against a set of criteria discussed in the following subsections. We randomly selected 103 document pairs for manual inspection. The sample size was calculated according to a 95% confidence level and an at most 10% confidence interval.

[13] We use the Java Wikipedia Library (Zesch et al., 2008) to convert each snapshot into a database that allows structured access to several aspects of categories, articles, sections etc.

5.1 Parallelness

The number of the correctly identified parallel document pairs was obviously critical in this particular evaluation setting. We focused on the precision of the pair detector module, since it is not feasible to count how many pairs were missed. In the subset examined, 94 and 4 document pairs were judged as parallel and not parallel, respectively. The other 5 pairs were considered borderline cases, where more than 20% of the sentences in one document were translated in the other. Since about 95% of the crawled data are of good or sufficiently good quality, this shows that they are usable for further processing, e.g. for sentence alignment.

5.2 Domain specificity

We next evaluated how many documents in the selected data fit the targeted domain in both the IT and the DE partitions. The overall precision was about 77%, with 79 IT documents and 80 DE documents found relevant to the narrow domain chosen for evaluation.

Reported results on text-to-topic classification sometimes score higher; however they neglect a critical factor of influence, namely the distance between training and prediction datasets. In the "real world", scores between 75% and 85% are realistic to assume. It should be mentioned that the precision of the topic classifier strongly depends on the quality of the seed terms: by inspecting results, modifying the seed term list and re-crawling, results could easily be improved further.

5.3 Language identification

Since the language identifier is applied on every paragraph of the main content of each web page, we examined how many of the paragraphs have been marked correctly. Overall, 5223 and 4814 paragraphs of IT and DE documents were checked and only 13 and 65 wrong assignments were found, respectively.

Most errors (about 80%) were found in a single document with a lot of tokens denoting chemical substances that seem to confuse the language identifier. When excluding this document, figures rise to 99,67% and 99,95% for the DE and IT partitions, respectively. The rest of the errors mainly occurred in paragraphs containing sentences in different languages.

EN: 547	DE: 255	EL: 22	ES: 293	FR: 286	IT: 143	PT: 164
Gardenia	Gardenien	Γαρδένια	Gardenia	Gardénia	Gardenia	Gardenia
Calendula	Ringelblumen	Καλέντουλα	Calendula	Calendula	Calendula	Calendula
Lilium	Lilien	Κρίνο	Lilium	Lys	Lilium	Lírio
Peony	Pfingstrosen	Παιώνια	Paeoniaceae	Pivoine	Paeonia	Paeoniaceae
Tulip	Tulpen	Τουλίπα	Tulipa	Tulipe	Tulipa	Tulipa
Flower	Blüte	Άνθος	Flor	Fleur	Fiore	Flor
Crocus	Krokusse	Κρόκος	Crocus	Crocus	Crocus	Crocus
Anemone	Windröschen	Ανεμώνη	Anemone	Anémone	Anemone	Anemone

Table 1: Sample seed terms for the "Flowers" domain in 7 languages, collected automatically from multilingual editions of Wikipedia. The header of the table refers to the total terms collected for each language.

Wikipedia article	Seed URL	WebSite	Langs
EN: Omphalodes_verna	http://goo.gl/msyIc	http://www.luontoportti.com	de,en,es,fr
ES: Tropaeolum	http://goo.gl/Ec5uK	http://www.chileflora.com	de,en,es
EN: Erythronium americanum	http://goo.gl/nEP2L	http://wildaboutgardening.org	en,fr
DE: Nickendes_Leimkraut	http://goo.gl/nuHNe	http://www.wildblumen.at	de,en,pt
DE: Titanenwurz	http://goo.gl/rLl9W	http://www.wilhelma.de	de,en

Table 2: Automatically detected web sites with multilingual content related to the "Flowers" domain. Column 1 presents the original LANG.wikipedia.org article from which the (shortened for readability purposes) seed URLs in column 2 were extracted. The seed URLs led to the 3rd column web sites, in which content in the languages of the 4th column was found.

5.4 Boilerplate removal

For this evaluation aspect, we evaluated how many "good" paragraphs were judged to be boilerplate, and how many "bad" paragraphs were missed. We examined 23178 and 23176 paragraphs of IT and DE documents and found 2326 and 2591 errors with an overall error rate around 10%. It should be noted that different strategies for boilerplate removal can be followed. One "classical" option is to remove everything that does not belong to the text, i.e. headers, advertisements etc. that "frame" real content. Another option is to attempt to remove everything which is irrelevant for MT sentence alignment; this goes beyond the first approach as it also removes short textual chunks, copyright disclaimers, etc. Most of the errors reported here were mainly due to this difference; i.e. they were paragraphs that were deemed not usable for MT alignment.

6 Conclusions and future work

In this paper we described and evaluated ILSP-FC, a system for mining domain-specific monolingual and bilingual corpora from the web. The system is available as open-source and is modular in the sense that each of its components can be easily substituted with similar software performing the same functionalities. The crawler can also be tested via web services that allow the user to perform experiments without the need to install it.

We have already used the crawler in producing monolingual and parallel corpora and other derivative resources. Evaluation has shown that the system can be used effectively in collecting resources of high quality, provided that the user can initialize it with lists of seed terms and URLs that can be easily found on the web. For domains for which no similar lists are available, we presented ongoing work for bootstrapping them from multilingual editions of Wikipedia. Future work includes evaluation and improvement of the bootstrapping component, more sophisticated methods for text classification, and grouping of collected data based on genre.

Acknowledgments

Work by the first two authors was partially funded by the European Union QTLaunchPad (FP7, Grant 296347) and Abu-MaTran (FP7-People-IAPP, Grant 324414) projects. An initial version of this work was produced during the EU Panacea project (FP7-ICT, Grant 248064).

References

Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. 2004. Applying support vector machines to imbalanced datasets. In *In Proceedings of the 15th European Conference on Machine Learning (ECML*, pages 39--50.

José João Almeida and Alberto Simões. 2010. Automatic parallel corpora and bilingual terminology extraction from parallel websites. In *3rd Workshop on Building and Using Comparable Corpora* .

Ethem Alpaydin. 2010. *Introduction to Machine Learning*. The MIT Press, 2nd edition.

Luciano Barbosa, Vivek Kumar Rangarajan Sridhar, Mahsa Yarmohammadi, and Srinivas Bangalore. 2012. Harvesting parallel text in multiple languages with limited supervision. In *COLING*, pages 201--214.

Marco Baroni, Adam Kilgarriff, Jan Pomikálek, and Pavel Rychlý. 2006. WebBootCaT: Instant Domain-Specific Corpora to Support Human Translators. In *Proceedings of the 11th Annual Conference of EAMT*, pages 47--252, Norway.

Marco Baroni, Francis Chantree, Adam Kilgarriff, and Serge Sharoff. 2008. Cleaneval: a competition for cleaning web pages. In *LREC'08*.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209--226.

Adrian W. Bowman and Adelchi Azzalini. 1997. *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, volume 18. Oxford University Press.

Jisong Chen, Rowena Chau, and Chung-Hsing Yeh. 2004. Discovering parallel text from the World Wide Web. In *Proceedings of ACSW Frontiers '04*, volume 32, pages 157--161, Darlinghurst, Australia.

Alain Désilets, Benoit Farley, Marta Stojanovic, and Geneviève Patenaude. 2008. WeBiText: Building Large Heterogeneous Translation Memories from Parallel Web Content. In *Proceedings of Translating and the Computer (30)*, London, UK.

Miquel Esplà-Gomis and Mikel L. Forcada. 2010. Combining Content-Based and URL-Based Heuristics to Harvest Aligned Bitexts from Multilingual Sites with Bitextor. *The Prague Bulletin of Mathematical Linguistics*, 93:77--86.

Gumwon Hong, Chi-Ho Li, Ming Zhou, and Hae-Chang Rim. 2010. An empirical study on web mining of parallel data. In *Proceedings of the 23rd COLING*, pages 474--482.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333--348.

Jian-Yun Nie, Michel Simard, Pierre Isabelle, and Richard Durand. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval*, pages 74--81, New York.

Christopher Olston and Marc Najork. 2010. Web crawling. *Found. Trends Inf. Retr.*, 4(3):175--246.

Pavel Pecina, Antonio Toral, Vassilis Papavassiliou, Prokopis Prokopidis, and Josef van Genabith. 2012. Domain adaptation of statistical machine translation using web-crawled resources: a case study. In *Proceedings of the 16th Annual Conference of EAMT*, pages 145--152, Trento, Italy.

Xiaoguang Qi and Brian D. Davison. 2009. Web page classification: Features and algorithms. *ACM Computing Surveys*, 41:11--31.

Philip Resnik and Noah A. Smith. 2003. The Web as a parallel corpus. *Computational Linguistics*, 29:349--380.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A dom tree alignment model for mining parallel data from the web. In *COLING/ACL-2006*, pages 489--496.

Miroslav Spousta, Michal Marek, and Pavel Pecina. 2008. Victor: the Web-Page Cleaning Tool. In *Proceedings of the 4th Web as Corpus Workshop - Can we beat Google?*, pages 12--17, Marrakech.

Padmini Srinivasan, Filippo Menczer, and Gautam Pant. 2005. A General Evaluation Framework for Topical Crawlers. *Information Retrieval*, 8:417--447.

Martin Theobald, Jonathan Siddharth, and Andreas Paepcke. 2008. Spotsigs: robust and efficient near duplicate detection in large web collections. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, pages 563--570.

Masao Utiyama, Daisuke Kawahara, Keiji Yasuda, and Eiichiro Sumita. 2009. Mining parallel texts from mixed-language web pages. In *MT Summit*.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech.

Ying Zhang, Ke Wu, Jianfeng Gao, and Phil Vines. 2006. Automatic Acquisition of Chinese-English Parallel Corpus from the Web. In *Proceedings of the 28th European Conference on Information Retrieval*, pages 420--431.

Building basic vocabulary across 40 languages

Judit Ács **Katalin Pajkossy** **András Kornai**
HAS Computer and Automation Research Institute
H-1111 Kende u 13-17, Budapest
{judit.acs,pajkossy,kornai}@sztaki.mta.hu

Abstract

The paper explores the options for building bilingual dictionaries by automated methods. We define the notion 'basic vocabulary' and investigate how well the conceptual units that make up this language-independent vocabulary are covered by language-specific bindings in 40 languages.

Introduction

Globalization increasingly brings languages in contact. At the time of the pioneering IBM work on the Hansard corpus (Brown et al., 1990), only two decades ago, there was no need for a Basque-Chinese dictionary, but today there is (Saralegi et al., 2012). While the methods for building dictionaries from parallel corpora are now mature (Melamed, 2000), there is a dearth of bilingual or even monolingual material (Zséder et al., 2012), hence the increased interest in comparable corpora.

Once we find bilingual speakers capable of carrying out a manual evaluation of representative samples, it is relatively easy to measure the precision of a dictionary built by automatic methods. But measuring recall remains a challenge, for if there existed a high quality machine-readable dictionary (MRD) to measure against, building a new one would largely be pointless, except perhaps as a means of engineering around copyright restrictions. We could measure recall against Wiktionary, but of course this is a moving target, and more importantly, the coverage across language pairs is extremely uneven.

What we need is a standardized vocabulary resource that is equally applicable to all language pairs. In this paper we describe our work toward creating such a resource by extending the *4lang* conceptual dictionary (Kornai and Makrai, 2013) to the top 40 languages (by Wikipedia size) using a variety of methods. Since some of the resources studied here are not available for the initial list of 40 languages, we extended the original list to 50 languages so as to guarantee at least 40 languages for every method. Throughout the paper, results are provided for all 50 languages, indicating missing data as needed.

Section 1 outlines the approach taken toward defining the basic vocabulary and translational equivalence. Section 2 describes how Wiktionary itself measures up against the 4lang resource directly and after triangulation across language pairs. Section 2.3 and Section 2.4 deals with extraction from multiply parallel and near-parallel corpora, and Section 3 offers some conclusions.

1 Basic vocabulary

The idea that there is a *basic* vocabulary composed of a few hundred or at most a few thousand elements has a long history going back to the Renaissance – for a summary, see Eco (1995). The first modern efforts in this direction are Thorndike's (1921) *Word Book,* based entirely on frequency counts (combining TF and DF measures), and Ogden's (1944) *Basic English,* based primarily on considerations of definability. Both had lasting impact, with Thorndike's approach forming the basis of much subsequent work on readability (Klare 1974, Kanungo and Orr 2009) and Ogden's forming the basis of the Simple English Wikipedia[1]. An important landmark is the Swadesh (1950) list, which puts special emphasis on cross-linguistic definability, as its primary goal is to support glottochronological studies.

Until the advent of large MRDs, the frequency-based method was much easier to follow, and Thorndike himself has extended his original list of ten thousand words to twenty thousand (Thorndike

[1] http://simple.wikipedia.org

Proceedings of the 6th Workshop on Building and Using Comparable Corpora, pages 52–58,
Sofia, Bulgaria, August 8, 2013.