Appendix A: MT evaluation

Domain Adaptation of Statistical Machine Translation using Web-Crawled Resources: A Case Study

Pavel Pecina, Antonio Toral, Josef van Genabith
School of Computing
Dublin City Universiy
Dublin 9, Ireland
{ppecina,atoral,josef}@computing.dcu.ie

Vassilis Papavassiliou, Prokopis Prokopidis
Institute for Language & Speech Processing
Artemidos 6 & Epidavrou
151 25 Maroussi, Greece
{vpapa,prokopis}@ilsp.gr

Abstract

In this research, we tackle the problem of domain adaptation of Statistical Machine Translation by exploiting domain-specific data acquired by domain-focused web-crawling. We design and empirically evaluate a procedure for automatic acquisition of both monolingual and parallel data and their exploitation for system training, tuning, and testing in a phrase-based Statistical Machine Translation framework. We present a strategy for using such resources depending on their availability and quantity supported by results of a large-scale evaluation carried out for the domains of Natural Environment and Labour Legislation, and two language pairs: English–French and English–Greek. The average observed improvement of BLEU is substantial at 49.5%.

1 Introduction

Recent advances of Statistical Machine Translation (SMT) have improved Machine Translation (MT) quality to such an extent that it can be successfully used in industrial processes (Flournoy and Duran, 2009). However, this mostly happens in very specific domains for which ample training data is available (Wu et al., 2008). Using in-domain data for training has a substantial effect on the final translation quality: SMT, as any other machine-learning application, is not guaranteed to perform optimally if the data for training and testing are not identically (and independently) distributed, which is often the case in practice. The main problem is usually vocabulary coverage: specific domain texts typically contain a substantial amount of special vocabulary that is not likely to be found in texts from other domains (Banerjee et al., 2010). Additional problems can be caused by divergence in style or genre, where the difference is not only in lexis but also in other linguistic aspects, such as grammar.

In order to achieve optimal performance, an SMT system should be trained on data from the same domain, genre, and style as the data is applied to. For many domains, though, in-domain data of a size sufficient to train a full SMT system is difficult to find. Recent experiments have shown that even small amounts of such data can be used to adapt an existing (general-domain) system to the particular domain of interest (Koehn et al., 2007). Sometimes appropriate sources of such data are in the form of existing in-house databases and translation memories (He et al., 2010). An alternative option is to exploit the constantly growing amount of text available on the web, although acquiring data of a sufficient quality and quantity from the web is a complicated process involving several critical steps (crawling, cleaning, etc.).

In this research, we present a strategy for automatic web-crawling and cleaning of domain-specific data with only minimal manual intervention, based on freely available tools deployed as web services. One advantage of this approach is that chaining of services enables building dynamic, flexible workflows which can always be improved by integration of new services and/or old legacy systems that may run on different technological platforms. Moreover the user does not have to deal with technical issues regarding the tools, such as their installation, configuration or maintenance. Further, our exhaustive experiments, carried out for the Natural Environment (*env*) and Labour Legislation (*lab*) domains and English–French (*EN–FR*) and English–Greek (*EN–EL*) language pairs (both directions), demonstrate how the crawled data improves translation quality and lead to interesting observations.

After an overview of related work, we discuss the possibility of adapting a general-domain SMT system by using various types of in-domain data. Then, we present our web-crawling procedure for monolingual and parallel data followed by a de-

scription of a series of experiments exploiting the data we acquired. Finally, we report on the results and conclude with recommendations for similar attempts to domain adaptation in SMT.

2 Related work and state of the art

2.1 Web crawling for textual data

Web crawlers travel the web by extracting links of already fetched web pages and by adding them to the list of pages to be visited. The selection of the next link to be followed is a key challenge for the evolution of thw crawl and is tied to the goal of the crawler. For example, a crawler that aims to index the Web may not order links, while a focused crawler that aspires to build domain-specific web collections (Qin and Chen, 2005) may use a relevance score as a ranking measure.

Several algorithms have been exploited for selecting the most promising links. The Best-First algorithm (Cho et al., 1998) sorts the links with respect to their relevance scores and selects a predefined amount of them as the seeds for the next crawling cycle. The PageRank (Brin and Page, 1998) algorithm exploits the "popularity" of a web page, i.e. the probability that a random crawler will visit that page at any given time, instead of its relevance. Menczer and Belew (2000) proposed an adaptive population of agents, called InfoSpiders, and searched for pages relevant to a domain using evolving query vectors and Neural Nets to decide which links to follow. Hybrid models and modifications of these crawling strategies have also been proposed (Gao et al., 2010; Dziwiński and Rutkowska, 2008) with the aim of reaching relevant pages rapidly. A general framework to fairly evaluate focused crawling algorithms under a number of performance metrics is proposed by Srinivasan et al. (2005).

2.2 Domain-focused web crawling

Apart from the crawling algorithm, classification of web content as relevant or not also affects the acquisition of domain-specific resources, on the assumption that relevant pages are more likely to contain links to more pages in the same domain. Qi and Davison (2009) review features and algorithms used in web page classification. In most of the algorithms reviewed, on-page features (i.e. textual content and HTML tags) are used to construct a corresponding feature vector (Golub and Ardö, 2005). Then, several machine-learning approaches such as SVMs, decision trees, and neural networks are employed (Yu et al., 2004). Many algorithms exploit additional information contained in web pages, including HTML tags, hyperlinks, and anchor text. Other methods adopt the assumption that neighbouring pages are likely to be on the same topic (Menczer, 2005).

2.3 Web-crawling for parallel data

Compared to crawling for monolingual data, acquisition of parallel data from the web is more challenging. Even though there are many websites with pairs of pages that are translations of each other, detection of such sites and identification of the pairs is far from straightforward.

Considering the Web as a parallel corpus, Resnik and Smith (2003) proposed the STRAND system, in which they used the AltaVista search engine to search for multilingual websites and examined the similarity of the HTML structures of the fetched web pages in order to identify pairs of potentially parallel pages. Besides structure similarity, similar 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) adopted a naive aligner in order to estimate the content similarity of candidate parallel web pages. Esplà-Gomis and Forcada (2010) proposed Bitextor, a system that combines language identification with shallow features (file size, text length, tag structure, and list of numbers in a web page) to mine parallel documents from multilingual web sites.

2.4 Domain adaptation in SMT

The first attempt towards domain adaptation in SMT was made by Langlais (2002) who integrated in-domain lexicons into the translation model. Eck et al. (2004) presented a language model adaptation technique applying an information retrieval approach based on selecting similar sentences from available training data. Hildebrand et al. (2005) applied the same approach on the translation model. Wu and Wang (2004) and Wu et al. (2005) proposed an alignment adaptation approach to improve domain-specific

language pair (L1–L2)	dom	set	source	sentences	L1 tokens / vocabulary		L2 tokens / vocabulary	
English–French	*gen*	train	Europarl 5	1,725,096	47,956,886	73,645	53,262,628	103,436
		dev	WPT 2005	2,000	58,655	5,734	67,295	6,913
		test	WPT 2005	2,000	57,951	5,649	66,200	6,876
English–Greek	*gen*	train	Europarl 5	964,242	27,446,726	61,497	27,537,853	173,435
		dev	WPT 2005	2,000	58,655	5,734	63,349	9,191
		test	WPT 2005	2,000	57,951	5,649	62,332	9,037

Table 1: Statistics of the general-domain data sets obtained from the Europarl corpus and WPT 2005 workshop.

word alignment. Munteanu and Marcu (2005) automatically extracted in-domain bilingual sentence pairs from large comparable (non-parallel) corpora to enlarge the in-domain bilingual corpus. Koehn and Schroeder (2007) integrated in--domain and out-of-domain language models as log-linear features in the Moses (Koehn et al., 2007) PB-SMT system with multiple decoding paths for combining multiple domain translation tables. Nakov (2008) combined in-domain translation and reordering models with out-of-domain models into Moses. Finch and Sumita (2008) employed a probabilistic mixture model combining two models for questions and declarative sentences with a general model. They used a probabilistic classifier to determine a vector of probability representing class membership.

In general, all approaches to domain adaptation of SMT depend on the availability of domain--specific data. If the data is available, it can be directly used to improve components of the MT system.If the data is not available, it can be extracted from a pool of texts from different domains (Eck et al., 2004; Hildebrand et al., 2005) or even from the web, which is also the case in our work.

3 Resources and their acquisition

In this section, we review the existing resources we used for training the general-domain systems and present the data acquisition procedures.

3.1 Existing general domain data

For the baseline, a general-domain system, we decided to exploit the widely used data provided by the organizers of the SMT workshops (WPT 2005 – WMT 2010): the Europarl parallel corpus (Koehn, 2005) as training data for translation and language models, and WPT 2005 test sets as the development and test data for general-domain parameter optimization and testing, respectively.

Europarl is extracted from the proceedings of the European Parliament. For practical reasons we consider this corpus to contain general--domain texts. Version 5, released in 2010, includes texts in 11 European languages including all languages of our interest (see Table 1). Note that the amount of parallel data for *EN–EL* is only about half of what is available for *EN–FR*. Furthermore, Greek morphology is more complex than French morphology so the Greek vocabulary size (we count unique lowercased alphabetical tokens) is much larger than the French one.

The WPT 2005 development and test sets are 2,000 sentence pairs each, available in the same languages as Europarl provided by the WPT 2005 organizers for the translation shared task. Later WMT test sets do not include Greek data.

3.2 In-domain data acquisition

A required resource for in-domain data acquisition is the definition of the targeted domain. Since we did not possess training data for the targeted domains and languages, we represented each domain as a list of weighted terms, following the approach by Ardö and Golub (2007). This method does not require any domain expertise since such terms are generally available on-line. To this end, we selected *EN*, *FR*, and *EL* terms (both single and multi-word entries) from the Natural Environment and Employment and Working Conditions domains of the EuroVoc thesaurus v4.3 (EuroVoc, 2011). Each entry was manually assigned a weight indicating the term's domain relevance, with higher values denoting more relevant terms. Alternatively, if in-domain data was available, the *tf-idf* weight could be used as a measure of term importance to eliminate the manual intervention.

3.2.1 Web-crawling for monolingual data

To acquire monolingual corpora, we implemented a focused monolingual crawler that adopts a distributed computing architecture based on Bixo (2011), an open source web mining toolkit that runs on top of Hadoop (2011).Heritrix (2011) web crawlers.

		initial phase				main phase						
language	**dom**	**sites**	**pages stored / sampled / acc**			**sites**	**pages visited / stored**	**(Δ%) /**	**dedup**	**(Δ%)**	***t* (h)**	
English	*env*	146	505	224	92.9	3,181	90,240	34,572	*38.3*	28,071	*18.8*	47
	lab	150	461	215	91.6	1,614	121,895	22,281	*18.3*	15,197	*31.8*	50
French	*env*	106	543	232	95.7	2,016	160,059	35,488	*22.2*	23,514	*33.7*	67
	lab	64	839	268	98.1	1,404	186,748	45,660	*27.2*	26,675	*41.6*	72
Greek	*env*	112	524	227	97.4	1,104	113,737	31,524	*27.7*	16,073	*49.0*	48
	lab	117	481	219	88.1	660	97,847	19,474	*19.9*	7,124	*63.4*	38
Average					***94.0***				***25.6***		***39.7***	

Table 2: Statistics from the initial (focused on domain-classification accuracy estimation) and main phases of crawling monolingual data: *stored* refers to the *visited* pages classified as in-domain, *dedup* refers to the pages after near-duplicate removal, *time* is the total duration (in hours), *acc* is accuracy estimated on the *sample* pages.

language	**dom**	**paragraphs all /**	**clean**	**(Δ%) /**	**unique**	**(Δ%)**	**sentences**	**tokens**	**vocabulary**
English	*env*	5,841,059	1,088,660	*18.6*	693,971	*11.9*	1,700,436	44,853,229	225,650
	lab	3,447,451	896,369	*26.0*	609,696	*17.7*	1,407,448	43,726,781	136,678
French	*env*	4,440,033	1,069,889	*24.1*	666,553	*15.0*	1,235,107	42,780,009	246,177
	lab	5,623,427	1,382,420	*24.6*	822,201	*14.6*	1,232,707	46,992,912	180,628
Greek	*env*	3,023,295	672,763	*22.3*	352,017	*11.6*	655,353	20,253,160	324,544
	lab	2,176,571	521,109	*23.9*	284,872	*13.1*	521,358	15,583,737	273,602
Average				***23.3***		***14.0***			

Table 3: Statistics from the cleaning stage of the monolingual data acquisition procedure and of the final data set: *clean* refers to paragraphs classified as non-boilerplate, *unique* refers to paragraphs kept after duplicate removal.

To initialize the crawler, we constructed lists of seed URLs that were, for the *env* domain, selected from relevant lists in the Open Directory Project (2011). For *lab* seed lists were generated from queries for random combinations of terms using the WebBootCat toolkit (Baroni et al., 2006).

The Best-first algorithm was adopted for crawl evolution in our implementation since this strategy is considered the baseline for almost all the relevant works. Each visited page is normalized to UTF-8 and its language is identified using the n-gram based method included in the Apache Tika toolkit. If the page is in the targeted language, it is compared to the topic definition and a page relevance score p is calculated as proposed by Ardö and Golub (2007). If this score is higher than a predefined threshold the page is classified as relevant to the domain. Then the links of each source page are extracted and the surrounding text of each link is located. A link relevance score l influenced by the source web page relevance score and the estimated relevance of the link's surrounding text is calculated as $l = p/N + \sum_{i=1}^{M} n_i \cdot w_i$, where N is the amount of links originating form the source page, M is the amount of terms in the topic definition, n_i denotes the number of occurrences of the i-th term in the surrounding text and w_i is the weight of the i-th term. This formulation of the link score was inspired by the conclusion of Cho et al. (1998), who stated that using a similarity metric that considers the content of anchors leads to some extent of differentiation among out--links and forces the crawler to visit relevant web pages earlier. New links are merged with the unvisited ones and sorted by their scores so the most promising links are selected for the next cycle.

Further processing steps during crawling include boilerplate detection and language identification. For the first we used a modified version of Boilerpipe (Kohlschütter et al., 2010) that also segments text in paragraphs exploiting HTML tags. We then applied a language identifier on each paragraph to check whether it is in the targeted language. Paragraphs classified to be in another language or detected as boilerplate were labeled and later filtered out.

In order to estimate the crawler's accuracy in acquiring in-domain resources we ran initial crawls in *EN*, *FR*, and *EL* for the *env* and *lab* domains and manually checked a sample of the acquired documents for domain relevance (see columns 3–6 in Table 2). Then we repeated the crawls to aquire larger collections (see columns 7–14). Near-duplicates were detected and removed by employing the deduplication strategy included in the Nutch framework.

language pair	dom	sites	docs	sents all / paired	(Δ%)	/ good	(Δ%)	/ unique	(Δ%)	/ sample / corrected	
English–French	*env*	6	559	19,042	14,881	*78.1*	14,079	*73.9*	13,840	*72.7*	3,600	3,392
	lab	4	900	35,870	31,541	*87.9*	27,601	*76.9*	23,861	*66.5*	3,600	3,411
English–Greek	*env*	14	288	17,033	14,846	*87.2*	14,028	*82.4*	13,253	*77.8*	3,600	3,000
	lab	7	203	13,169	11,006	*83.6*	9,904	*75.2*	9,764	*74.1*	2,700	2,506
Average						***84.2***		***77.1***		***72.8***		

Table 4: Statistics from the parallel data acquisition procedure: total document pairs (*docs*), source side sentences (*sents all*), aligned sentences pairs (*paired*), those of sufficient translation quality (*good*); after duplicate removal (*unique*); sentences randomly selected for manual correction (*sample*) and those really corrected (*corrected*).

The ratio of pages classified as in-domain is 25.6% in average (column 11 in Table 2) which is similar to the results achieved by Srinivasan et al. (2005) and Dorado (2008). The relatively high percentages of documents removed during deduplication (column 13 in Table 2) are in accordance with Baroni et al.'s (2009) observation that during building of the Wacky corpora the amount of documents was reduced by more than 50% after deduplication. Another observation is that the percentages of duplicates for the *lab* domain are much higher than the ones for the *env* for each language. This is explained by the fact that the web pages related to *lab* are mainly legal documents or press releases replicated on many websites.

Final processing of the monolingual data was performed on paragraphs identified by Boilerpipe using HTML tags. The statistics from this phase are presented in Table 3. Firstly, we discarded all paragraphs in languages different then the targeted ones and those classified as boilerplate, which reduced their total amount to 23.3% in average. Removal of duplicate paragraphs then reduced the total number of paragraphs to 14.0% in average. However, most of the removed paragraphs were very short chunks of text (such as navigation links, etc.). In term of tokens, the reduction is only to 50.6%. The last three columns in Table 3 refer to the final monolingual data sets used for training language models. For *EN* and *FR*, we acquired about 45 million tokens for each domain; for *EL*, which is less frequent on the web, we obtained only about 15–20 million tokens.

3.2.2 Web-crawling for parallel data

Some steps involved in parallel data acquisition (including normalization, language identification, cleaning, and deduplication) were discussed in the previous subsection as a part of the monolingual data acquisition. To guide the focused bilingual crawler we used sets of bilingual topic definitions. In order to construct the list of seed URLs we selected web pages that were collected during the monolingual crawls and originated from in--domain multilingual web sites. Since it is likely that these multilingual sites contain parallel documents, we initialize the crawler with these seed URLs and force the crawler to follow only links internal to these sites. In this scenario, the selection of a crawling strategy that prioritizes the links to be visited is not of great importance. This observation motivated us to select a Breadth--First algorithm (the simplest crawling algorithm that considers extracted links as a first-in-first-out queue) that interacts with a text classifier. After downloading in-domain pages from the selected web sites, we employed Bitextor to identify pairs of documents that could be considered parallel.

3.2.3 Parallel sentence extraction

After identification of parallel documents, the next steps aimed at extraction of parallel sentences. For each document pair free of boilerplate paragraphs, we applied the following steps: identification of sentence boundaries by the Europarl sentence splitter, tokenization by the Europarl tokenizer, and sentence alignment by Hunalign (Varga et al., 2005). Hunalign implements a heuristic, language-independent method for identification of parallel sentences in parallel texts which can be improved by providing an external bilingual dictionary of word forms. Without having such dictionaries for *EN–FR* and *EN–EL* at hand, we opted to realign data in these languages from Europarl by Hunalign and used the dictionaries produced by this tool.

For each sentence pair identified as parallel, Hunalign provides a confidence score which reflects the level of parallelness, We manually investigated a sample of sentence pairs extracted by Hunalign from the pool data (about 50 sentence pairs for each language pair and domain), by relying on the judgment of native speakers, and estimated that sentence pairs with a score above 0.4

are of a good translation quality. In the next step, we kept sentence pairs with 1:1 alignment only (one sentence on each side) and removed those with scores below this threshold. Finally, we also removed duplicate sentence pairs.

The statistics from the parallel data acquisition procedure are displayed in Table 4. on average, 85% of source sentences extracted from the parallel documents were aligned in the 1:1 fashion, 10% of them were then removed due to low translation quality, and after discarding duplicate sentences pairs we ended up with 72% of the original source sentences aligned to their target sides.

3.2.4 Manual correction of parallel sentences

The translation quality of the parallel sentences obtained by the procedure described above is not guaranteed in any sense. Tuning the procedure and focusing on high-quality translations is possible but leads to a trade-off between quality and quantity. For translation model training, high translation quality of the data is not as essential as for testing. Bad phrase pairs can be removed from the translation tables based on their low translation probabilities. However, a development set containing sentence pairs which are not good translations of each other might lead to sub-optimal values of model weights which would harm system performance. If such sentence pairs are used in the test set, the evaluation would clearly be unreliable.

In order to create reliable test (and development) sets for each language pair and domain, we performed the following low-cost procedure. From the data obtained by the steps described in the previous section, we selected a random sample of 3,600 sentence pairs (2,700 for *EN–EL* in the *lab* domain, for which less data was available) and asked native speakers to check and correct them. The task consisted of checking that the sentence pairs belonged to the right domain, the sentences within a sentence pair were equivalent in terms of content, and the translation quality was adequate and (if needed) correcting it.

Our goal was to obtain at least 3,000 correct sentence pairs for each domain and language pair; thus the correctors did not have to correct every sentence pair. They were allowed to skip (remove) those sentence pairs which were misaligned. In addition, we asked them to remove those sentence pairs that were obviously from a very different domain (despite being correct translations). The number of corrected sentence pairs is presented in the last column of Table 4.

	EN–EL / env	***EN–FR / lab***
1. perfect translation	53.49	72.23
2. minor corrections done	34.15	21.99
3. major corrections needed	3.00	0.33
4. misaligned sentence pair	5.09	1.58
5. wrong domain	4.28	3.86

Table 5: Statistics of manual correction of parallel data.

dom	**set**	**sents**	**L1 toks**	**/ voc**	**L2 toks**	**/ voc**
EN–FR *env*	train	10,240	300,760	10,963	362,899	14,209
	dev	1,392	41,382	4,660	49,657	5,542
	dev*	1,458	42,414	4,754	50,965	5,700
	test	2,000	58,865	5,483	70,740	6,617
EN–FR *lab*	train	20,261	709,893	12,746	836,634	17,139
	dev	1,411	52,156	4,478	61,191	5,535
	dev*	1,498	54,024	4,706	63,519	5,832
	test	2,000	71,688	5,277	84,397	6,630
EN–EL *env*	train	9,653	240,822	10,932	267,742	20,185
	dev	1,000	27,865	3,586	30,510	5,467
	dev*	1,134	32,588	3,967	35,446	6,137
	test	2,000	58,073	4,893	63,551	8,229
EN–EL *lab*	train	7,064	233,145	7,136	244,396	14,456
	dev	506	15,129	2,227	16,089	3,333
	dev*	547	17,027	2,386	18,172	3,620
	test	2,000	62,953	4,022	66,770	7,056

Table 6: Statistics of the in-domain parallel data sets obtained by web-crawling and manual correction.

According to the human judgments, 53–72% of sentence pairs were accurate translations, 22–34% needed only minor corrections, 1–3% would require major corrections (which was not necessary, as the accurate sentence pairs together with those requiring minor corrections were enough to reach our goal of at least 3,000 sentence pairs), 2–5% of sentence pairs were misaligned and would have had to be translated completely, and about 4% of sentence pairs were from a different domain (though correct translations), see Table 5.

Further, we selected 2,000 pairs from the corrected sentences for the test set and left the remaining part for the development set. The parallel sentences which were not selected for corrections were used as training sets. Further statistics of these sets are given in Table 6. The correctors confirmed that the manual corrections were about 5–10 times faster than translating the sentences from scratch, so this can be viewed as low-cost method for acquiring in-domain test sets for MT.

4 Domain adaptation experiments

In this section, we describe our SMT system and present experiments that exloit all the acquired in-domain data and evaluated in eight different scenarios involving two domains (*env*, *lab*), two language pairs (*EN–FR*, *EN–EL*) in both translation directions. Our primary evaluation measure is BLEU (Papineni et al., 2002). For detailed analysis we also present NIST (Doddington, 2002) and METEOR (Banerjee and Lavie, 2005) in Table 8.

4.1 System description

Our SMT system is MaTrEx, a combination-based multi-engine architecture (Penkale et al., 2010) exploiting aspects of both the Example-based Machine Translation and SMT paradigms. In this work, we only exploit the SMT phrase-based component which is based on Moses.

For training the baseline MT system, training data is tokenized and lowercased using the standard Europarl tools. The original (non-lowercased) versions of the target sides of the parallel data are kept for training the Moses recaser. The lowercased versions of the target sides are used for training an interpolated 5-gram language model with Kneser-Ney discounting using the SRILM toolkit (Stolcke, 2002). Translation models are trained on the relevant parts of the Europarl corpus, lowercased and filtered on sentence level; we kept all sentence pairs having less than 100 words on each side and with length ratio within the interval $\langle 0.11,9.0 \rangle$. Minimum error rate training (Och, 2003, MERT) is used to optimize the model parameters on the development set.

For decoding, test sentences are tokenized, lowercased, and translated by the tuned system. Letter casing is then reconstructed by the recaser and extra blank spaces in the tokenized text are removed in order to produce human-readable text.

4.2 Using out-of-domain test data

A number of previous experiments (Wu et al., 2008; Banerjee et al., 2010, etc.) showed significant degradation of translation quality if an SMT system was applied to out-of-domain data. In order to verify this observation we trained and tuned our system on general-domain data and compared its performance on in-domain (*gen*) and out-of-domain (*env*, *lab*) test sets (the systems are denoted as *vx* and *v0*, respectivelly). The average decrease in BLEU score is 44.3%: while on in-domain test sets we observe scores in the interval 42.24–57.00, the scores on the out-of-domain test sets are in the range 20.20–31.79. This is obviously caused by the divergence of training and test data; also the OOV rate increased from 0.25% to 0.90% (see columns *vx* and *v0* in Table 7).

4.3 Using in-domain development data

Optimization of parameters of the SMT log-linear models is known to have a big influence on the performance. The first step towards domain adaptation of a general-domain system it to use in-domain development data. Such data usually comprises of a small set of parallel sentences which are repeatedly translated while the model parameters are adjusted towards their optimal values. The minimum number of development sentences is not strictly given. The only requirement is that the optimization procedure (MERT in our case) must converge, which might not happen if the set is too small. By using the parallel data acquisition procedure described in Sections 3.2, we acquired development sets (506–1,411 sentence pairs in each) which proved to be very beneficial for parameter tunning: compared to the baseline systems trained and tuned on general-domain data only (denoted as *v0*), systems trained on general-domain data and tuned on in-domain data (denoted as *v1*) improved BLEU scores by 25.5% on average. Taking into account that the development sets contain only several hundreds of parallel sentences each, such improvement is remarkable (compare columns *v0* and *v1* in Table 7).

4.4 Corrected vs. raw development data

We put a certain effort into the manual correction of the development data used in the previous experiments. In order to justify the need for development data of good translation quality we took the baseline systems (trained on general-domain data) and tuned them using the raw, uncorrected sentence pairs. This raw development data (denoted by * in Table 6) contains not only the sentences with imperfect translation, but also those which are misaligned and/or belong to other domains. As a consequence, the raw development sets contain 5–14% more sentences pairs than the corrected ones. Performance of the systems tuned using the raw development data is presented in column *v2* of Table 7. Surprisingly, in all scenar-

dir	dom	*vx*	/ OOV	dom	*v0*	/ OOV	*v1*	/Δ%	*v2*	/Δ%	*v3*	/Δ%	*v4*	/Δ%	*v5*	/Δ%	/ OOV
EN-FR gen	49.12	0.11	*env*	28.03	0.98	35.81	*27.8*	36.00	*28.4*	39.23	*40.0*	40.53	*44.6*	40.72	*45.3*	0.65
			lab	22.26	0.85	30.84	***38.5***	30.19	***35.6***	34.00	*52.7*	39.55	***77.7***	39.35	***76.8***	0.48
FR-EN gen	57.00	0.11	*env*	31.79	0.81	39.04	*22.8*	38.93	*22.5*	40.57	*27.6*	42.23	*32.8*	42.17	*32.7*	0.54
			lab	27.00	0.68	33.52	*24.2*	33.39	*23.7*	38.07	*41.0*	44.14	*63.5*	43.85	*62.4*	0.38
EN-EL gen	42.24	0.22	*env*	20.20	1.15	26.18	*29.6*	26.07	*29.1*	32.06	***58.7***	33.83	*67.5*	34.50	*70.8*	0.82
			lab	22.92	0.47	28.79	*25.6*	28.82	*25.7*	33.59	*46.6*	33.54	*46.3*	33.71	*47.1*	0.40
EL-EN gen	44.15	**0.56**	*env*	29.23	**1.53**	34.16	*16.9*	34.15	*16.8*	36.93	*26.3*	39.13	*33.9*	39.18	*34.0*	1.20
			lab	31.71	0.69	37.55	*18.4*	37.67	*18.8*	40.17	*26.7*	40.44	*27.5*	40.33	*27.2*	0.62
Average		**0.25**			**0.90**		***25.5***		***25.1***		***40.0***		***49.2***		***49.5***	**0.64**

Table 7: Results (BLEU scores) of domain adaptation of baseline general-domain systems (*v0*) by exploiting: corrected development data (*v1*), un-corrected development data (*v2*), monolingual training data (*v3*), parallel training data (*v4*), both monolingual and parallel training data (*v5*). *vx* refers to baseline systems applied to general domain test sets, *OOV* to out-of-vocabulary rates, and Δ% to relative improvement over the baseline systems (*v0*).

ios the difference compared to systems *v1* is not statistically significant (Koehn, 2004, p=0.05) and the average improvement over the baseline system *v0* is 25.1%, which is comparable to the score of 25.5% obtained by systems *v1*. This observation makes the correction of development data obtained by our procedure unnecessary and the raw data can be used for parameter tunning as it is.

4.5 Adding in-domain monolingual data

Improving an SMT system by adding in-domain monolingual training data cannot reduce the relatively high OOV rate observed when general--domain systems were applied on test sets from specific domains. However, such data can improve the language models and contribute to better estimations of probabilities of n-grams consisting of known words. To verify this hypothesis, we trained systems (denoted as *v3*) on general--domain parallel training data, in-domain development data (corrected), and a concatenation of general-domain and in-domain monolingual data described in Section 3.2.1 (comprising 15–45 million words). Compared to the systems *v1*, the BLEU scores were improved by additional 14.5% absolute in average. In comparison with the baseline systems *v0*, the total increase of BLEU is 40.0% in average. The most substantial improvement over the system *v1* is achieved for translations to Greek (23.0% for *env*, and 16.2% for *lab*) despite the smallest size of the monolingual data acquired for this language (see Table 3) which is probably due to the complex Greek morphology.

4.6 Adding in-domain parallel training data

Parallel data is essential for building translation models of SMT systems. While a good language model can improve an SMT system by preferring better translation options in given contexts, it has no effect if the translation model offers no translation at all, which is the case for OOV words (and longer phrases too). In the next experiment, we analyze the effect of using additional in-domain parallel training data acquired as described in Section 3.2.3 (comprising 7–20 thousand sentence pairs). First, we trained systems (denoted as *v4*) on a concatenation of general--domain and in-domain parallel training data, in--domain development data, and a general-domain monolingual data only (no extra in-domain monolingual data) which outperformed the previous systems (*v3*) by additional 9.2% absolute in average. The total improvement of BLEU with respect to the baseline (*v0*) is then an impressive 49.2% in average. In certain scenarios, the overall improvement was above 70%. To provide a complete picture we also trained fully adapted systems (denoted as *v5*) using both general-domain and in--domain sets of parallel and monolingual data and tuned on the corrected in-domain development sets with the following observations: in most scenarios the difference of results of these systems compared to systems *v4* are not statistically significant (p=0.05). The average relative improvement over the baseline (*v0*) is 49.5%, which is almost identical to 49.2% from the previous experiment (*v4*). In practice, this means that using additional monolingual in-domain data on top of the in-domain parallel data has no effect on the translation quality at all. Although additional experiments are needed to verify whether larger monolingual data could bring any additional improvement or not, it seems that parallel data is more important in this context.

		Natural Environment								Labour Legislation							
	sys	**BLEU**	**Δ%**	**NIST**	**Δ%**	**MET**	**Δ%**	**WER**	**Δ%**	**BLEU**	**Δ%**	**NIST**	**Δ%**	**MET**	**Δ%**	**WER**	**Δ%**
English–French	*v0*	28.03	*0.0*	7.03	*0.0*	63.32	*0.0*	63.70	*0.0*	22.26	*0.0*	6.27	*0.0*	56.73	*0.0*	69.93	*0.0*
	v1	35.81	*27.7*	8.10	*15.2*	68.44	*8.0*	53.78	*-15.5*	30.84	*38.5*	7.42	*18.3*	62.94	*10.9*	57.99	*-17.0*
	v2	36.00	*28.4*	8.16	*16.0*	68.27	*7.8*	53.24	*-16.4*	30.19	*35.6*	7.31	*16.5*	62.84	*10.7*	59.11	*-15.4*
	v3	39.23	*39.9*	8.43	*19.9*	70.35	*11.1*	51.34	*-19.4*	34.00	*52.7*	7.68	*22.4*	65.56	*15.5*	57.06	*-18.4*
	v4	40.53	*44.6*	8.61	*22.4*	71.10	*12.2*	50.04	*-21.4*	39.55	*77.6*	8.37	*33.4*	69.82	*23.0*	52.04	*-25.5*
	v5	40.72	*45.2*	8.63	*22.7*	71.23	*12.4*	49.92	*-21.6*	39.35	*76.7*	8.34	*33.0*	69.79	*23.0*	52.29	*-25.2*
French–English	*v0*	31.79	*0.0*	7.77	*0.0*	66.25	*0.0*	57.09	*0.0*	27.00	*0.0*	7.07	*0.0*	59.90	*0.0*	61.57	*0.0*
	v1	39.04	*22.8*	8.75	*12.6*	69.17	*4.4*	48.26	*-15.4*	33.52	*24.1*	7.98	*12.8*	63.70	*6.3*	53.39	*-13.2*
	v2	38.93	*22.4*	8.74	*12.4*	69.06	*4.2*	48.36	*-15.2*	33.39	*23.6*	7.97	*12.7*	63.53	*6.0*	53.46	*-13.1*
	v3	40.57	*27.6*	8.90	*14.5*	70.23	*6.0*	47.19	*-17.3*	38.07	*41.0*	8.47	*19.8*	66.88	*11.6*	50.35	*-18.2*
	v4	42.23	*32.8*	9.09	*16.9*	71.40	*7.7*	46.07	*-19.3*	44.14	*63.4*	9.22	*30.4*	71.24	*18.9*	45.49	*-26.1*
	v5	42.17	*32.6*	9.09	*16.9*	71.32	*7.6*	46.05	*-19.3*	43.85	*62.4*	9.17	*29.7*	71.07	*18.6*	45.81	*-25.6*
English–Greek	*v0*	20.20	*0.0*	5.73	*0.0*	82.81	*0.0*	67.83	*0.0*	22.92	*0.0*	5.93	*0.0*	87.27	*0.0*	65.88	*0.0*
	v1	26.18	*29.6*	6.57	*14.6*	84.19	*1.6*	60.80	*-10.3*	28.79	*25.6*	6.80	*14.6*	87.91	*0.7*	58.20	*-11.6*
	v2	26.07	*29.0*	6.60	*15.1*	84.70	*2.2*	60.14	*-11.3*	28.82	*25.7*	6.79	*14.5*	88.18	*1.0*	58.32	*-11.4*
	v3	32.06	*58.7*	7.24	*26.3*	84.52	*2.0*	56.68	*-16.4*	33.59	*46.5*	7.36	*24.1*	88.34	*1.2*	54.71	*-16.9*
	v4	33.83	*67.4*	7.63	*33.1*	86.10	*3.9*	53.47	*-21.1*	33.54	*46.3*	7.34	*23.7*	89.55	*2.6*	54.68	*-17.0*
	v5	34.50	*70.7*	7.57	*32.1*	85.91	*3.7*	54.16	*-20.1*	33.71	*47.0*	7.34	*23.7*	89.42	*2.4*	54.71	*-16.9*
Greek–English	*v0*	29.23	*0.0*	7.50	*0.0*	60.57	*0.0*	54.69	*0.0*	31.71	*0.0*	7.76	*0.0*	62.42	*0.0*	52.34	*0.0*
	v1	34.16	*16.8*	8.01	*6.8*	64.98	*7.2*	51.15	*-6.4*	37.55	*18.4*	8.28	*6.7*	67.36	*7.9*	49.02	*-6.3*
	v2	34.15	*16.8*	7.96	*6.1*	64.99	*7.3*	51.29	*-6.2*	37.67	*18.8*	8.34	*7.4*	67.31	*7.8*	48.64	*-7.0*
	v3	36.93	*26.3*	8.27	*10.2*	66.60	*9.9*	49.40	*-9.6*	40.17	*26.6*	8.58	*10.5*	68.67	*10.0*	47.03	*-10.1*
	v4	39.13	*33.8*	8.55	*14.0*	68.24	*12.6*	47.94	*-12.3*	40.44	*27.5*	8.61	*10.9*	68.91	*10.4*	46.78	*-10.6*
	v5	39.18	*34.0*	8.54	*13.8*	68.19	*12.5*	47.94	*-12.3*	40.33	*27.1*	8.60	*10.8*	68.83	*10.2*	47.00	*-10.2*

Table 8: Overview of domain adaptation results. With the exception of NIST, all scores are percentages; MET denotes METEOR, system identifiers refer to those in Table 7, and Δ to relative improvement over the systems (*v0*).

5 Conclusions

In this research, we presented two methods for the acquisition of domain-specific monolingual and parallel data from the web. These employ existing open-source tools for normalization, language identification, cleaning, deduplication, and parallel sentence extraction and are implemented as easy to use webservices ready to be employed in industrial scenarios (e.g. in a translation company providing services supported by MT). These methods applied to acquire monolingual and parallel data for two language pairs and two domains with the only required manual intervention for the domain definitions and the list of seed URLs.

The acquired resources were then successfully used to adapt general-domain SMT systems to the new domains. The average relative improvement of BLEU scores achieved in eight scenarios was a substantial 49.5%. Based on our experiments we made the following observations: even small amounts of in-domain parallel data is more important for translation quality than large amounts of in-domain monolingual data. As few as 500–1,000 sentence pairs can be used as development data with expected 25% relative improvement of BLEU scores. Importantly, this data does not have to be corrected, MERT seems quite tolerant to imperfect translation of development sets (to a certain extent, of course). Additional parallel data can be used to improve translation models: 7,000–20,000 sentences pairs in our experiments increased our BLUE scores by other 25 absolute in average. If such data is not available, a general--domain system can benefit from using additional in-domain monolingual data, however quite large amounts (tens of million words) are necessary to obtain a moderate improvement.

Our domain adaptation of SMT is based on the trivial concatenation of general-domain and domain-specific data, however future work may reveal that other, more sophisticated, approaches are more appropriate.

References

Anders Ardö and Koraljka Golub. 2007. Focused crawler software package. Technical report, Lund University, Department of Information Technology.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan.

Pratyush Banerjee, Jinhua Du, Baoli Li, Sudip Naskar, Andy Way, and Josef van Genabith. 2010. Combining Multi-Domain Statistical Machine Translation Models using Automatic Classifiers. In *AMTA 2010: The Ninth Conference of the Association for Machine Translation in the Americas*, pages 141–150.

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 the European Association for Machine Translation*, pages 47–252, Norway.

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.

Bixo. 2011. An open source web mining toolkit. http://openbixo.org/.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN Syst.*, 30:107–117, April.

Jisong Chen, Rowena Chau, and Chung-Hsing Yeh. 2004. Discovering parallel text from the World Wide Web. In *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, volume 32 of *ACSW Frontiers '04*, pages 157–161, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. 1998. Efficient crawling through URL ordering. *Comput. Netw. ISDN Syst.*, 30:161–172, April.

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.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, pages 138–145, San Diego, California.

Ignacio Garcia Dorado. 2008. Focused Crawling: Algorithm Survey and new Approaches with a Manual Analysis. Master's thesis, Department of Electro and Information Technology, Lund University.

Piotr Dziwiński and Danuta Rutkowska. 2008. Ant Focused Crawling Algorithm. In *Proceedings of the 9th international conference on Artificial Intelligence and Soft Computing*, ICAISC '08, pages 1018–1028, Berlin, Heidelberg. Springer-Verlag.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2004. Language Model Adaptation for Statistical Machine Translation based on Information Retrieval. In *International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

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.

EuroVoc. 2011. The EU's multilingual thesaurus. http://eurovoc.europa.eu/.

Andrew Finch and Eiichiro Sumita. 2008. Dynamic model interpolation for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 208–215, Columbus, Ohio, USA.

Raymond Flournoy and Christine Duran. 2009. Machine translation and document localization at Adobe: from pilot to production. In *MT Summit XII: proceedings of the twelfth Machine Translation Summit*, pages 425–428.

Zhaoqiong Gao, Yajun Du, Liangzhong Yi, Yuekui Yang, and Qiangqiang Peng. 2010. Focused Web Crawling Based on Incremental Learning. *Journal of Computational Information Systems*, 6:9–16.

Koraljka Golub and Anders Ardö. 2005. Importance of HTML Structural Elements and Metadata in Automated Subject Classification. In *Proceedings of ECDL 2005, 9th European Conference*, pages 368–378, Vienna, Austria. Springer.

Hadoop. 2011. A distributed computing platform. http://hadoop.apache.org.

Yifan He, Yanjun Ma, Johann Roturier, Andy Way, and Josef van Genabith. 2010. Improving the Post-Editing Experience Using Translation Recommendation: A User Study. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, pages 247–256.

Heritrix. 2011. Internet Archive web crawler. http://crawler.archive.org/.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation*, pages 133–142, Budapest, Hungary.

Wu Hua, Wang Haifeng, and Liu Zhanyi. 2005. Alignment model adaptation for domain-specific word alignment. In *43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 467–474, Ann Arbor, Michigan, USA.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 224–227, Prague, Czech Republic.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Prague, Czech Republic.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate detection using shallow text features. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 441–450, New York.

Philippe Langlais. 2002. Improving a general-purpose Statistical Translation Engine by terminological lexicons. In *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology - Volume 14*, pages 1–7, Taipei, Taiwan.

Filippo Menczer and Richard K. Belew. 2000. Adaptive Retrieval Agents: Internalizing Local Contextand Scaling up to the Web. *Machine Learning*, 39:203–242, May.

Filippo Menczer. 2005. Mapping the Semantics of Web Text and Links. *IEEE Internet Computing*, 9:27–36, May.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *Comput. Linguist.*, 31:477–504.

Preslav Nakov. 2008. Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 147–150, Columbus, Ohio, USA.

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, NY, USA. ACM.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *41st Annual Meeting on Association for Computational Linguistics*, ACL '03, pages 160–167, Sapporo, Japan.

ODP. 2011. Open Directory Project. http://dmoz.org.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, USA.

Sergio Penkale, Rejwanul Haque, Sandipan Dandapat, Pratyush Banerjee, Ankit K. Srivastava, Jinhua Du, Pavel Pecina, Sudip Kumar Naskar, Mikel L. Forcada, and Andy Way. 2010. MaTrEx: the DCU MT system for WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 143–148, Uppsala, Sweden.

Xiaoguang Qi and Brian D. Davison. 2009. Web page classification: Features and algorithms. *ACM Computing Surveys*, 41:12:1–12:31, February.

Jialun Qin and Hsinchun Chen. 2005. Using Genetic Algorithm in Building Domain-Specific Collections: An Experiment in the Nanotechnology Domain. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 102.2–, Washington, DC, USA. IEEE Computer Society.

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

Padmini Srinivasan, Filippo Menczer, and Gautam Pant. 2005. A General Evaluation Framework for Topical Crawlers. *Inf. Retr.*, 8:417–447, May.

Andreas Stolcke. 2002. SRILM-an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, pages 257–286, Denver, Colorado, USA.

Dániel Varga, Laszlo Németh, Péter Halácsy, András Kornai, Viktor Trón, , and Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Recent Advances in Natural Language Processing (RANLP 2005)*, pages 590–596.

Hua Wu and Haifeng Wang. 2004. Improving domain-specific word alignment with a general bilingual corpus. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas*, pages 262–271, Washington, DC.

Hua Wu, Haifeng Wang, and Chengqing Zong. 2008. Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 993–1000.

Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. 2004. PEBL: Web Page Classification without Negative Examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81.

Ying Zhang, Ke Wu, Jianfeng Gao, and Phil Vines. 2006. Automatic Acquisition of Chinese-English Parallel Corpus from the Web. In *Proceedings of ECIR-06, 28th European Conference on Information Retrieval*, pages 420–431.