PaCo^2: A Fully Automated tool for gathering Parallel Corpora from the Web

Iñaki San Vicente, Iker Manterola

R&D Elhuyar Foundation
{i.sanvicente, i.manterola}@elhuyar.com

Abstract

The importance of parallel corpora in the NLP field is well known. This paper presents a tool that can build parallel corpora given just a seed word list and a pair of languages. Our approach is similar to others proposed in the literature, but introduces a new phase to the process. While most of the systems leave the task of finding websites containing parallel content up to the user, PaCo^2 (Parallel Corpora Collector) takes care of that as well. The tool is language independent as far as possible, and adapting the system to work with new languages is fairly straightforward. Evaluation of the different modules has been carried out for Basque-Spanish, Spanish-English and Portuguese-English language pairs. Even though there is still room for improvement, results are positive. Results show that the corpora created have very good quality translations units, and the quality is maintained for the various language pairs. Details of the corpora created up until now are also provided.

Keywords: Parallel Corpora, Corpus Building, Multilingual Resources

1. Introduction

It is well-known how important Parallel Corpora are in the NLP field and beyond. Maybe the best known application area would be MT, where statistical systems greatly depend on such resources. But we can not forget areas such as terminology mining or human translation, where parallel corpora are widely used. Unfortunately, a major handicap this kind of resource has, is its scarceness. Even if we turn to a pair of major languages, it is difficult to find domain specific parallel corpora with an amount of words enough to train an MT system or to run a terminology extraction application. In the last decade, researchers have increasingly turned their eyes to comparable corpora. Comparable corpora are easier to obtain, but the results obtained with such corpora do not achieve those of parallel corpora. So, when available parallel corpora are still preferred.

Since the late 1990's, there have been some proposals to automatically gather parallel corpora from the Web. Most of them (Yang and Li, 2003) (Fry, 2005) (Espla-Gomis, 2009) focus on finding documents which are translations of each other (also called bitexts) on a previously specified website. PaCo^2 includes an initial step whereby sites containing bitexts are automatically identified. The tool is designed to be as language independent as possible and it is fairly easy to adapt it to new languages. Our intention is to make this tool freely available to the research community.

The paper is organized as follows: section 2. provides a brief review of the state of the art in the field. Next sections present the architecture of our tool and discuss its different components. We describe the evaluation carried out to test our tool in section 4., and comment the results we obtained. Finally we draw some conclusions and we point out possible future lines of work.

2. State Of The Art

Much research has been done finding parallel texts or sentences from the Web. Some focused on extracting parallel sentences out of comparable corpora (Fung and Cheung, 2004; Munteanu and Marcu, 2005). Machine Learning classifiers and bilingual dictionaries are used to pair sentences of the different languages, which need language specific resources. Although positive results are achieved using this approach, the proportion of parallel content in comparable data is low. Smith et al. (2010) noticed that Wikipedia has a great quantity of parallel content linked at document level, but then again, the parallel content greatly decreases in the case of not major languages, as shown for Bulgarian in their article.

Utiyama et al. (2009) propose to extract parallel sentences from mixed language web pages, that is, pages that have content in more than one language. The scarcity of this type of web pages is a major handicap for this method.

The most common approach has been to look for bitexts in websites containing parallel content and then align the bitexts at sentence level. Identifying parallel websites can be a part of the process (Nie et al., 1999; Resnik and Smith, 2003; Zhang et al., 2006), or the source can be previously fixed (Chen et al., 2004; Espla-Gomis, 2009). Document-level alignment is typically done by using different features of the candidate documents, such as document URL similarity (Nie et al., 1999), inter-language links and HTML structure (Resnik and Smith, 2003; Shi et al., 2006). Some authors compare the content of the documents, using bilingual dictionaries to break the language barrier (Fukushima et al., 2006), or extracting named entities that are language independent (Nadeau and Foster, 2004).

PaCo^2 can be considered to be from this last group. This approach presents two problems: on the one hand, some of the aforementioned features are dependant on the structure of the website (e.g., document URLs, inter-language links). However, when applicable these features achieve a very high precision and recall making them very attractive. On the other hand, the main bottleneck of this approach is the large number of document comparisons that need to be made. Combining the different features offers a solution to both problems, because it provides a fast method for

websites that fully comply with some specific features, and reduce the number of comparisons in the other cases.

3. System Architecture

PaCo[2] is designed as a pipeline comprising three main phases. The user can launch the tool in any phase of the process, according to his/her needs. The first phase searches the Web and looks for websites that contain bilingual content. In the second phase parallel documents are detected for each of the sites gathered in the previous phase. The third and last phase aligns detected bitexts at sentence level, and builds the final parallel corpus. The different phases are discussed in the next sections.

3.1. Bilingual Website Detection

The most novel feature of our tool is its capacity to find parallel sites by itself. Although earlier approaches already incorporated such a feature (Nie et al., 1999), (Resnik and Smith, 2003), their technique relied on a search option provided by the Altavista search engine and which is not available anymore. Later approaches use previously identified sources (Nadeau and Foster, 2004) (Fry, 2005) (Chen et al., 2004). To our knowledge, the most similar proposal to ours is WPDE (Zhang et al., 2006); the main difference is that we use search engines to gather the candidate sites, while they use a fixed list to look for those candidates. It is important to automatically identify parallel websites. For most language pairs there is not enough material to build a fairly large parallel corpus if we only rely on a limited number of sources. Moreover, it is difficult to gather such sources manually. Turning to search engines allows us to get corpora as large as the Web can offer.

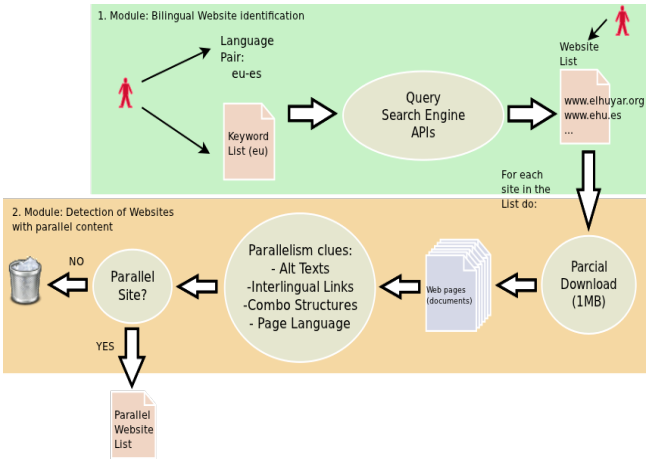

Figure 1: Bilingual detection phase is divided in two modules: candidate sites gathering (1) and parallel sites detection (2).

The first step for detecting bilingual sites is to gather possible candidates (see Figure 1). To do this, we start with a list of words in one language, preferably the one with less presence on the Web. Using random word combinations we query search engines, and gather the URLs returned. Search engines can be configured to look only for pages in a certain language. We make use of this feature when available, but other tweaks such as morphological query expansion (Leturia et al., 2008) can be used to ensure that the sites returned have content in a specific language and maximize the recall of this phase. This feature is implemented at the moment only for the Basque language. As for the word combinations sent to query the search engines, the system is prepared to combine words from two word lists, one including entities and the other including non-entity words. This implementation intends to address the matter of creating domain-specific corpora, although this paper is not focused on that issue. For the experiments carried out in this paper we used three-word combinations of non-entity keywords, extracted over a few hundred words list composed by lemmas. Nevertheless a list as short as 50 words could be enough.

The results returned by the search engines usually have a lot of not valid web domains. A preliminary filter is applied in order to discard *a priori* inadequate sites, such as gigantic sites, blog platforms, and social networks. Blog platforms mostly host personal blogs written in a single language. Accepting them at this stage would suppose little benefit in terms of recall, at the cost of a great effort for the next step of the process, as well as a probably precision decrease. Social networks or collaborative platforms (e.g. Wikipedia) have very few parallel documents compared with the amount of data they host, although they have parallel content. Probably the approach to extract parallel sentences from comparable corpora is more suitable in those cases than the one proposed in this paper.

The second step is to identify those websites which have parallel content (see Figure 1). To do this, we download a small part of each site (1 MB of data per site, only including HTML pages), and look for parallelism clues. For each web page in a site we look for hyperlinks that have anchor texts or ALT texts indicating languages (Zhang et al., 2006). We have also noticed that many sites use drop-down lists to connect with different translations of a page[1]. If a site has a minimum number of documents containing these clues, it is chosen as a bilingual site.

3.2. Bitext detection

In order to pair parallel texts from a specific website we built a module using state-of-the-art techniques. The module consists of various heuristics and statistical filters. PaCo[2] harvests all the n HTML documents from a website using Wget[2]. In the worst case, a document would be compared to all other documents, in other words, we would make $n(n-1)$ comparisons. A series of heuristics are applied in order to reduce as much as possible the number of comparisons the subsequent filters carry out (Espla-Gomis, 2009). Those heuristics are based on the language[3], file size and character length of the candidate documents. A document d_a in a language a is exclusively compared against those documents in language b which are

[1]It is usual to find drop-down lists that provide the translation of pages using services like Google Translate. We discard the sites that use such services.

[2]http://www.gnu.org/s/wget/

[3]Language identification is done using TextCat - http://www.let.rug.nl/vannoord/TextCat/

inside certain similarity thresholds with regard to the file size and character length of d_a.
The bitext detection module runs three major filters: link follower filter, URL pattern search, and a combination of an HTML structure filter and a content filter.
Before any comparison is done, all documents are preprocessed. Character encoding is standardized to utf-8 by means of BeautyfulSoup[4] , and documents are converted to raw text, maintaining their text structure. Although the original HTML documents are still used by the HTML structure filter, raw text format is mainly used throughout the process. No boilerplate removal is done. The reason for this is that we find that menus and other elements such as breadcrumbs do contain useful parallel information. In addition, content that remains unaltered regardless of the language of the document (e.g., copyright notes, contact information) is cleaned by the sentence alignment module (see section 3.3.).

3.2.1. Link follower filter

Documents in multilingual websites are often connected to their translations in other languages. Let us assume that a document in Basque (eu) d_{eu} contains a link to a document in Spanish (es) d_{es}. If the anchor text is a significant language mark *(spanish OR español OR es OR ...)* we follow that link to the translation candidate. However, in order to ensure that d_{es} and d_{eu} are bitexts we impose two restrictions:

- The connection must be bidirectional. d_{es} must also contain a language link *(Basque OR Euskara OR eu OR ...)* connecting to d_{eu}.
- d_{es} must be in the translation candidate set for d_{eu}.

3.2.2. URL pattern search

As other authors noticed, (Nie et al., 1999) (Resnik and Smith, 2003) bitexts often have similar file names and paths, only differing in some language marks. For example, the document d_{eu} in Basque and its translation $t(d_{eu}) = d_{es}$ have the URLs *"www.euskonews.com/0289zbk/elkarEU.html"* and *"www.euskonews.com/0289zbk/elkarES.html"*, respectively. For each pair of documents language marks are stripped from the URLs, and then the Longest Common Subsequence Ratio (LCSR) is calculated. If the score reaches a certain threshold the candidates are regarded as bitexts.

3.2.3. HTML structure and Content filters

The previous filters are highly dependant on the structure of the website. When that structure fits one of those filters, the system provides high quality bitexts within a short time. Unfortunately this is not always the case. When no heuristic is applicable, we propose that the HTML tag structure information (Resnik and Smith, 2003) be combined with a content similarity filter. We observed that HTML structure is not always able to pick out wrong candidates, due to many candidate web pages being almost identical, except for some minor changes.

As a solution we implemented a filter that calculates the similarity between two documents. Those pairs that reach a certain similarity threshold are regarded as bitexts. In order to avoid the language barrier when computing similarity we focus on extracting "universal entities", such as numbers, emails or dates (Nadeau and Foster, 2004). This allows us to keep the filter language independent. A document d is represented by a vector of entities sorted in order of appearance. The popular Ratcliff/Obershelp (Ratcliff et al., 1988) algorithm is used to compare documents in one language against their bitext candidates.
For each bitext candidate $d_{esi} \in t(d_{eu}) = \{d_{es1}, .., d_{esn}\}$ of a document d_{eu}, HTML structure similarity and content similarity are computed. Both results are weighted and those candidates that reach a certain threshold will be regarded as correct bitexts. If more than one candidate d_{esi} reach the threshold, we consider that the filter could not find the correct bitext, and all candidates are ruled out. This behavior gives priority to precision over recall.

3.3. Sentence Alignment

Sentence alignment is done by using the Hunalign[5] tool. We cannot guarantee that the bitexts we feed into Hunalign are totally parallel, and that is why, post-processing is also applied in order to clean wrong alignments and invalid translation units (TU).
First of all, TUs are sorted and duplicates are grouped. TUs that exclusively contain elements such as magnitudes, numbers, email addresses and/or URLs are removed. Language identification is done in order to ensure that the languages are correct for both source and target sentences that form a translation unit. If one of them is wrong, the TU is excluded.
The frequency of a TU is also provided. A TU repeated frequently over different domains has a high probability to be correct. If a TU appears frequently but in a single domain, it could be correct, but it could also be an alignment error. In this last case we accept the TU, as we have not enough evidence to reject it.
Lastly, there are several source segments s_{eu} which have multiple translation sentences $\{t_{es1}, ..t_{esm}\}$. All TUs $\{TU_1, .., TU_m\}$ where $TU_j = (s_{eu}, t_{esj})$ that include a unique source sentence s_{eu} are excluded if the group contains more than two elements ($m > 2$). As we can not discern which one is the correct TU, again, we act in favor of precision even if it means to lose some valid translation units.
Output is given as raw text or TMX format.

4. Evaluation and Results

Various corpora have been built using PaCo[2]. A Basque-Spanish parallel corpus is being constructed to be used in SMT. At the time of submitting this paper the corpus has 19.4M words (659,395 segments).
Other corpora include a Spanish-English corpus and a Portuguese-English one, both in the academic domain. Those corpora were built starting from manually selected sites, in order to gather content from a specific domain only.

[4] http://www.crummy.com/software/BeautifulSoup/

[5] http://mokk.bme.hu/resources/hunalign/

These corpora are part of another experiment, but they gave us the opportunity to test the tool with different languages. Table 1 shows statistics from the different phases of the corpus collecting process, until the final cleaned translation memory is created.

Corpus	#websites	#bitexts	#TUs	#words
eu-es	86	179,124	659,395	19,413,008
es-en	74	28,437	240,466	7,264,985
pt-en	43	1,569	29,983	872,877

Table 1: Statistics for the various corpora built.

All of the evaluation experiments have been carried out for the three aforementioned language pairs. Likewise, the results described from here on correspond to the three corpora mentioned above, except for the experiments in the 4.1. section, where the results for the Spanish-English corpus and a Portuguese-English language pairs are no related to the academic corpora described in this section.

4.1. Bilingual Website Detection

Experiments to measure the performance of the website detection phase were carried out for different language pairs. We measured the accuracy and productivity of this phase. In order to do that, we launched a set of 1,000 random queries to the Bing[6] search engine, and analyzed the results obtained. All language pairs ended obtaining candidates within 2 hours, and processing those candidates took 8 hours in the case of the Portuguese-English pair, and 30-40 hours in the other cases.

Corpus	#Autom. obtained candidates	#Autom. selected parallel sites	#Correct parallel sites	accuracy	productivity
eu-es	8,297	672	653	0.97	0.07
es-en	7,747	433	292	0.67	0.03
pt-en	1,716	92	72	0.78	0.04

Table 2: Results of the website Detection module.

Table 2 shows the results for the three language pairs. Results vary notably from one language pair to the others. We can see that the productivity is very low in all cases. This was to be expected since we are querying search engines using a single language, and the proportion of websites containing a specific language pair with respect to the rest of the web is low. Best results are obtained for the Basque-Spanish pair. The fact that Basque and Spanish are coexisting and co-official languages could be a reason why there is a higher density of parallel content in the WWW.
Accuracy is very high in the case of the Basque-Spanish language pair, but decreases greatly for other language pairs. An error analysis was conducted over the wrong candidates of Spanish and Portuguese. The conclusion is that many monolingual sites (specially blogs) include elements such as tags and term cloud, which can include misleading links. The fact that the Basque candidates do not suffer from this problem can be due to the preliminary filter being more accurately tuned for Basque Web. Nevertheless, the module needs to be improved in order to detect such false clues.
Lastly the number of results obtained for the Portuguese-English pair is surprisingly low. However, it must be noted that we were looking for European Portuguese domain candidates, and thus, candidates from Brazilian Portuguese domains were excluded. We analysed the results provided by the search engine, and if we had added those domains, we would have 7,241 candidates.

[6]http://www.bing.com

4.2. Bitext detection

For sites that implement interlingual links or URL patterns, this phase achieves nearly 100% precision with a good recall. In those sites the HTML/content filter is able to find very few new bitexts. For sites where the system can only use the HTML/content filter performance decreases in terms of precision, although a greater number of new bitexts are harvested.
We have evaluated the performance of this filter over a set of 9 randomly selected web domains, three for each language pair. In both cases the bitexts returned by the HTML/content filter were manually evaluated. Results in table 3 show that the HTML/content filter provides new bitexts, although the contribution varies from one language pair to the others. As expected, the accuracy of the new bitexts decreases notably compared to the values achieved by the previous filters.

Corpus	#bitexts found without HTML/ content	#bitexts found by HTML/ content	Accuracy without HTML/ content	Accuracy of HTML/ content
eu-es	2,059	143	0.99	0.7
es-en	509	576	1	0.64
pt-en	91	482	1	0.79

Table 3: Results of the bitext detection module.

Looking at the bitexts candidates evaluated, we noticed that the HTML/content filter module fails to rule out wrong bitext candidates when translation candidates are automatically created pages such as review forms or opinion forms. This kind of pages have similar contents and usually differs on some minimun number of words such as the name of a product or the title of a film. Those bitexts are considered correct only if the exact match is found, although it could be discussed if they are totally wrong, since the content is almost parallel.
Lastly, seeing the great variation in the results between different language pairs, the evaluated web domains were analysed. We observed that the nature of the websites is very different: they have very different number of documents, topics, etc.; some websites mainly have long documents while others mainly have documents with very few content such as tables and forms. That leads us the conclusion that deeper analysis is needed to determine the

reason behind the variability in the performance of the HTML/content filter.

4.3. Sentence Alignment

The quality of the sentence alignment phase is evaluated over two random sets of 1,000 TUs per language pair. The first set was extracted from the alignment produced by Hunalign, and is used as the baseline of the module. It must be noticed that the precision and recall of Hunalign is reported to be very high (Varga et al., 2005), but we expect it to be lower in this task, because the conditions are harder in our scenario. The bitexts we feed to Hunalign are not always parallel. The second test-set of TUs is extracted from the final parallel corpus. Each TU was evaluated once. Two annotators took part in this task. They are bilingual speakers of Basque and Spanish, with advanced knowledge of English and one of them has a medium knowledge of Portuguese. The annotators were asked to mark if the TU was undoubtedly correct (1), if translation was partial (2), if the overall meaning was correct but expressed with different sentences (3) or else if the translation was overall wrong (0).

Corpus		Correct	Partial	Correctly aligned	Total
eu-es	pre	0.82	0.04	0.02	0.88
	post	**0.88**	0.02	0.01	**0.91**
es-en	pre	0.82	0.06	0.01	0.88
	post	**0.87**	0.01	0.02	**0.91**
pt-en	pre	0.85	0.06	0.01	0.92
	post	**0.9**	0.01	0.02	**0.93**

Table 4: Results of the evaluation of the sentence alignment. *pre* rows represent results for the first test-set, while *post* rows represent results for the test-set derived from the final corpora.

As we can see in table 4, the post processing step we include after the initial alignment improves the accuracy of the TUs by a 5-6% for all of the language pairs evaluated. It could be doubtful whether segments marked as partial and correctly aligned should be included in the final corpora. The first group includes correct translations, but they are only partial. We could try to improve the alignment in order to correct those TUs. Segments on the second group have not been "literally" translated, one language may include more information than the other, or despite of the overall meaning being correct they have some kind of error (see table 5 for examples). Discarding this kind of TUs is beyond the reach of our tool at the moment. Nevertheless, if those two groups of TUs are considered correct, the overall accuracy of the corpora would rise up to 93%.

4.4. Time consumption

Regarding the time cost to create the corpora, it is very difficult to make an accurate estimation, as it depends on many parameters. One of the most time consuming tasks is downloading the web-domains, but the tool can not do much about it as it mainly depends on the Network. All of the experiments were carried out in a Linux server with 8 CPUs of 2.4 GHz speed and 16GB of RAM memory.

Es	En
De este modo, se orienta hacia la insercin laboral y la formacin de postgrado.	*Finally, upon successful completion of the degree, students may either begin professional practice or go on to postgraduate studies.*
Vas para acceder al Grado	*How to access:*
En la segunda planta hay varias habitaciones: la autoestima, las aptitudes personales y sociales, y el sentido del humor.	*The first floor has several rooms: self-esteem, personal and social skills and sense of humour.*

Table 5: Examples of correct alignments with doubtful translations. The first includes more information in one language than in the other. The last example has an error of meaning (English version says *"first floor"* while Spanish version says *"second floor"*).

The main bottleneck regarding the time, would be the bitext detection module. Its performance varies greatly depending on the structure and the size of the website it is processing. In our experiments, PaCo^2 is capable of processing a website with 100,000 documents in less than 48 hours if the link follower filter can be applied, but it can take up to 5 days if the URL pattern filter and HTML/content filter has to make many comparisons.

To give a general idea of the time needed for the whole process, the open-domain Basque-Spanish corpus, the largest corpus we have gathered up to now, took one month and a half of uninterrupted processing.

5. Conclusions

We have presented a tool for gathering parallel corpora from the web. The tool implements state of the art techniques for the task of finding bitexts in a web domain that contains parallel content. In addition, it is also capable of finding those web domains automatically. PaCo^2 has been successfully used for building corpora of various language pairs. The performance of the different stages of the process has been analysed, and results are promising.

Nevertheless, there is still room for improvement. Parallel website detection module should be improved in order to better adapt to new languages. In addition, the boundaries of the parallel website detection module are still unexplored. The performance of HTML/content filter needs to be analysed more deeply in order to improve its accuracy.

Using the corpora built with PaCo^2 in a real environment would allow us to measure the usefulness of our tool. We have two possible scenarios in mind: training of SMT systems and a terminology extraction task.

Finally, we have only superficially addressed the matter of creating domain specific parallel corpora. Our future work will also go in that direction.

6. Acknowledgements

This work has been partially founded by the Industry Department of the Basque Government under grants IE09-262 (Berbatek project) and SA-2010/00190 (AWPC project).

7. References

Jisong Chen, Rowena Chau, and Chung-Hsing Yeh. 2004. Discovering parallel text from the world wide web. In *The second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32*, pages 157–161.

M. Espla-Gomis. 2009. Bitextor, a free/open-source software to harvest translation memories from multilingual websites. In *Beyond Translation Memories Workshop (MT Summit XII)*.

J. Fry. 2005. Assembling a parallel corpus from RSS news feeds. In *MT Summit X*, page 59.

Ken'ichi Fukushima, Kenjiro Taura, and Takashi Chikayama. 2006. A fast and accurate method for detecting English-Japanese parallel texts. In *Workshop on Multilingual Language Resources and Interoperability*, pages 60–67, Sydney, Australia. Association for Computational Linguistics.

Pascale Fung and Percy Cheung. 2004. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

I. Leturia, A. Gurrutxaga, N. Areta, and E. Pociello. 2008. Analysis and performance of morphological query expansion and language-filtering words on basque web searching. In *Proceedings of the 6th edition of the LREC conference*.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, 31(4):477–504, December.

D. Nadeau and G. Foster. 2004. Real-time identification of parallel texts from bilingual news-feed. In *CLiNE2004*, pages 21–28.

J. Y Nie, M. Simard, P. Isabelle, and R. Durand. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In *SIGIR'99*, pages 74–81.

J.W. Ratcliff, D. Metzener, et al. 1988. Pattern matching: The gestalt approach. *Dr. Dobbs Journal*, 7:46.

P. Resnik and N. A Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):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 *21st COLING/44th ACL*, pages 489–496. Association for Computational Linguistics.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 403–411, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. Utiyama, D. Kawahara, K. Yasuda, and E. Sumita. 2009. Mining parallel texts from Mixed-Language web pages. *Proceedings of the XII Machine Translation Summit*.

D. Varga, L. Nmeth, P. Halcsy, A. Kornaia nd V. Trn, and V. Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of RANLP'05*, RANLP'05, pages 590–596.

C. C Yang and K. W Li. 2003. Automatic construction of English/Chinese parallel corpora. *Journal of the American Society for Information Science and Technology*, 54(8):730–742.

Y. Zhang, K. Wu, J. Gao, and P. Vines. 2006. Automatic acquisition of Chinese-English parallel corpus from the web. *Lecture Notes in Computer Science*, 3936:420.